Production Linux Clusters 2003

Architecture and System Software for Serious Computing

Peter Beckman, beckman@mcs.anl.gov, Argonne National Laboratory
Susan Coghlan, smc@mcs.anl.gov, Argonne National Laboratory
Rémy Evard, evard@mcs.anl.gov, Argonne National Laboratory
William Saphir, wcsaphir@lbl.gov, Lawrence Berkeley National Laboratory

November 16, 2003

Production Linux Clusters 2003

Architecture and System Software for Serious Computing

Peter Beckman, beckman@mcs.anl.gov, Argonne National Laboratory
Susan Coghlan, smc@mcs.anl.gov, Argonne National Laboratory
Rémy Evard, evard@mcs.anl.gov, Argonne National Laboratory
William Saphir, wcsaphir@lbl.gov, Lawrence Berkeley National Laboratory

Our Focus: Production Linux Clusters

Production

- Performance and uptime requirements.
- Many users.
- Many applications, and many types of applications.
- High-performance computing (HPC). Parallel programs.

Linux

- The most popular OS choice for cluster-based computing.

Clusters

- Cluster architecture systems, i.e. tightly-coupled systems supporting distributed memory computing.
- All scales of clusters, from 8 nodes to thousands.

This model is particularly applicable to the SC community.

Other cluster models (e.g. single application) are typically less complex. Many topics we cover will be applicable to these.

Slide 2

Why We're Here and Who We Are

- At SC99, Pete, Bill and Remy presented an earlier version of this tutorial, with the basic message that Linux clusters could be used for serious computing.
- Four years later, while Linux clusters have taken off, it's still a challenge to build and operate a production Linux cluster.
- Pete – Director of Engineering, TeraGrid, a grid of production Linux clusters. Formerly VP Engineering of Turbolinux.
- Susan – Lead systems engineer on Jazz, Argonne's 350-node terascale Linux cluster. Formerly managed LANL's ASCI computing systems.
- Remy – Lead architect on Chiba City (512 cpus), Jazz, and Argonne's co-site-lead in TeraGrid.
- Bill – Chief Architect for the NERSC high performance computing center. Previously Director of Engineering at Scale8.

Slide 3

Tutorial Organization

- The topics are organized in the order you would go through if you were to start with nothing (except budget) and end up with a working production cluster.
- The Schedule

8:30 – 10:00	Linux, Clusters, Early Decisions, Node Architectures
10:30 – 12:00	Networks, Storage, Purchasing and Installation
1:30 – 3:00	Administration, Resource Management, Applications
3:30 – 5:00	HPC Distributions, Production, Fitting It All Together

- In your notes:
  - A table of contents.
  - Appendix A – References.
  - Appendix B – The cluster setup checklist.

★ means "recommended"
● indicates a reference
✓ is a useful checklist

- Feel free to ask questions at any time.

Slide 4

Morning, Part I - Detailed Outline

- Introduction (these slides) – Remy
- Linux Clusters & HPC– Pete
  - Some important background information on Linux and Linux-based clusters.
- Cluster Architectures and Usage – Bill
  - How clusters are organized, used, and built, including important terminology and philosophy.
- Node Architectures – Susan and Pete
  - CPUs, motherboard issues, and all the other choices that you need to select the right type of node for a cluster.

Slide 5

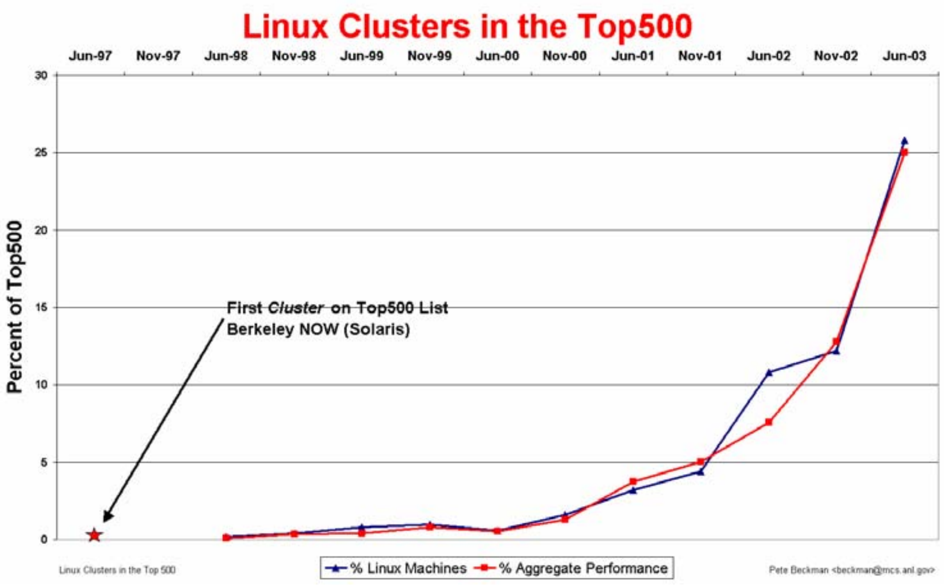
Linux Clusters & HPC

Slide 6

A Trip in the Way Back Machine….

- Top500 1993
  - Thinking Machines CM5, NEC SX-3, Intel Delta, Cray Y-MP
  - Shift Happens. MPPs compete with vector machines
- Beowulf project & Berkeley NOW project explore clustering
- People start asking questions…
- **1993**: Can a Linux cluster provide cost effective cycles for a single scientist?
- **1997**: Linux cluster wins Gordon Bell award for cost/performance
- **1998**: Linux cluster makes Top500, press calls it a "Supercomputer"
- **1999**: SC99 Tutorial on "Production Linux Clusters"
- **2000**: Can a Linux cluster provide a stable, multi-user HPC environment for a wide-variety of applications?
- **2003**: Linux clusters provide DOE & NSF centers production environments for thousands of users

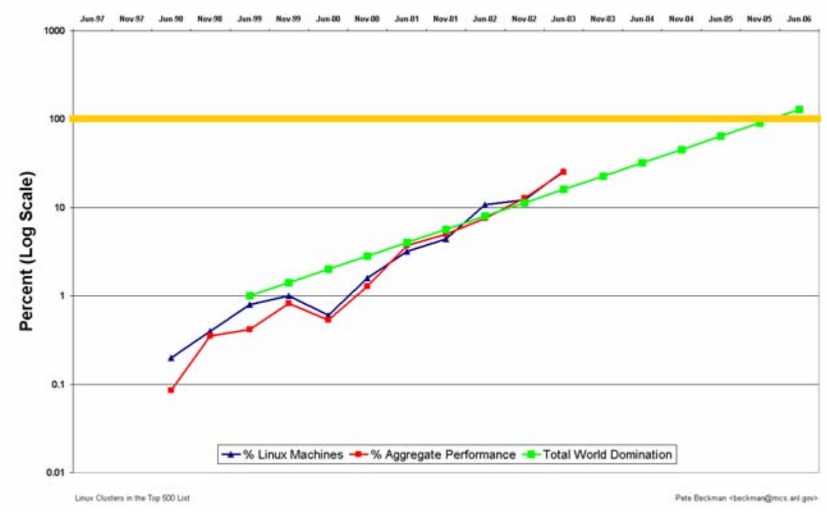
Slide 7



Slide 8

Predicting Future Market Share

How Long Until Total World Domination?



Slide 9

Welcome to the Age of Linux for HPC

- The adoption rate of Linux HPC is phenomenal!
  - Linux in the Top500 is doubling every 12 months
  - Linux adoption is not driven by bottom feeders
    - ***Adoption is actually faster at the ultra-scale!***
- Prediction: by 2005, most supercomputers will run Linux
- Adoption rate driven by several factors:
  - Linux software is stable, widely available commodity OS
    - Often the default platform for computer science research
  - Commodity hardware ROCKS!
  - Essentially no barrier to entry
  - Effort to learn programming paradigm, libs, devl env., and tools preserved across many orders of magnitude
  - Stable, complete, portable, middleware software stacks:
    - MPICH, MPI-IO, PBS, math libraries, etc

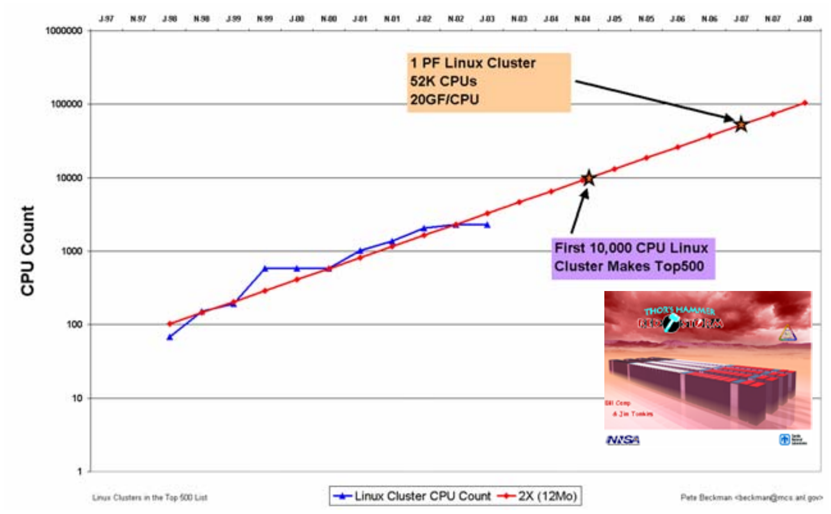
Slide 10

Could Linux Cluster Computing be the Bicycle for HPC?

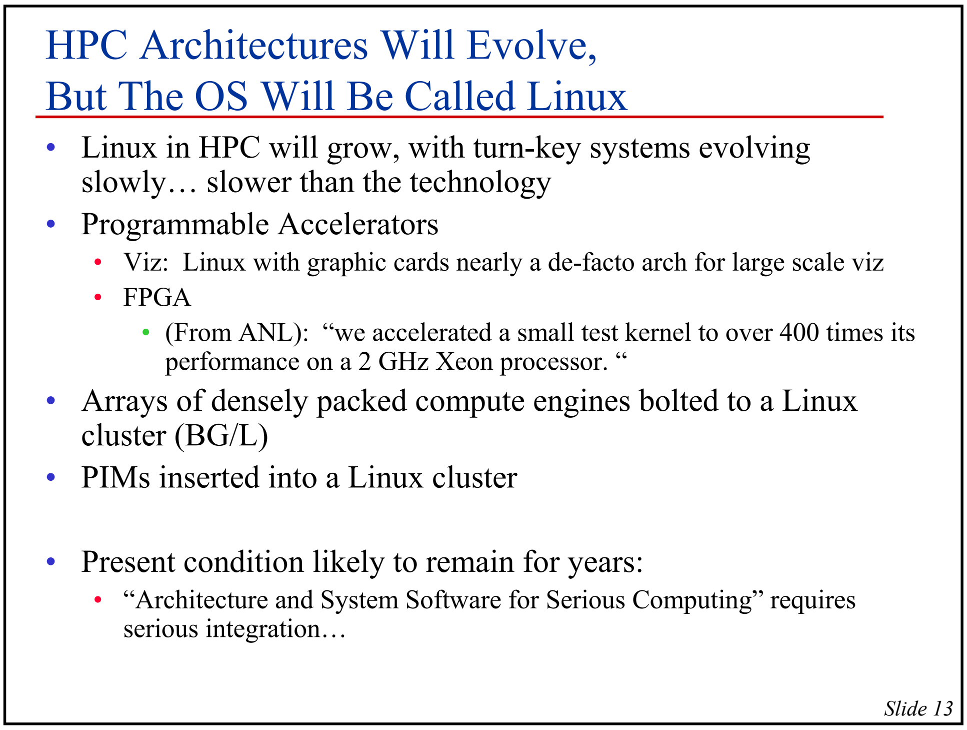
- Optimal design
  - Several initial design iterations followed by 100 yrs of stable form
  - Form will survive another 100 yrs
- Everyone can afford one
- Everyone can learn
- Parts are everywhere
- Operational knowledge ubiquitous

Slide 11

Linux Cluster Sizes: They are Getting Bigger Too



Slide 12

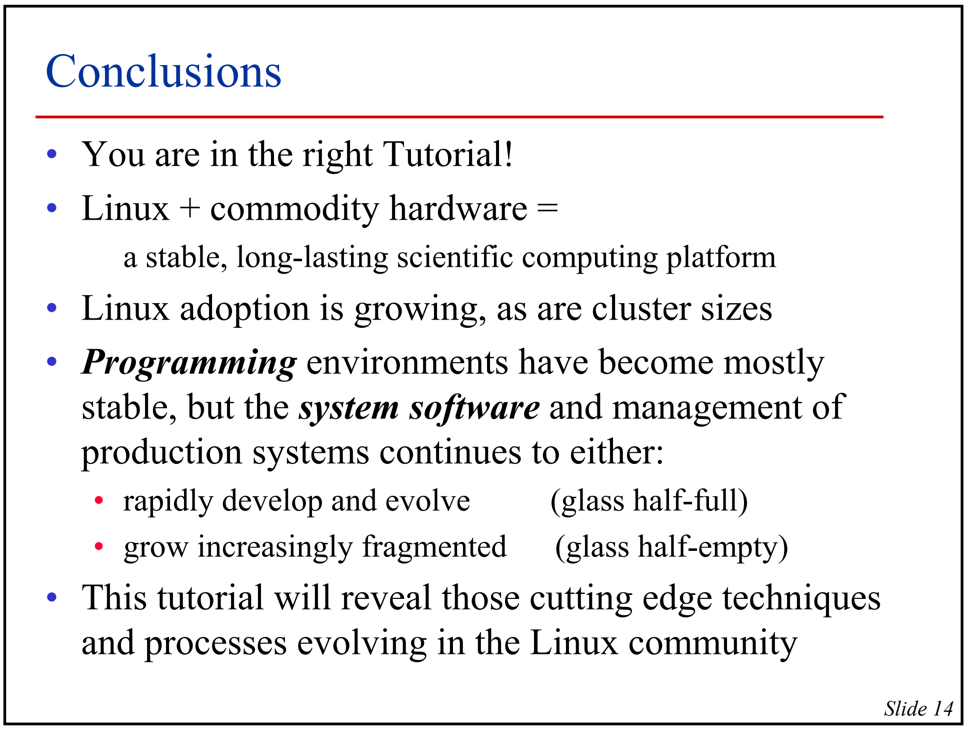


HPC Architectures Will Evolve, But The OS Will Be Called Linux

- Linux in HPC will grow, with turn-key systems evolving slowly… slower than the technology
- Programmable Accelerators
  - Viz: Linux with graphic cards nearly a de-facto arch for large scale viz
  - FPGA
    - (From ANL): "we accelerated a small test kernel to over 400 times its performance on a 2 GHz Xeon processor. "
- Arrays of densely packed compute engines bolted to a Linux cluster (BG/L)
- PIMs inserted into a Linux cluster

- Present condition likely to remain for years:
  - "Architecture and System Software for Serious Computing" requires serious integration…

Slide 13



Conclusions

- You are in the right Tutorial!
- Linux + commodity hardware =

  a stable, long-lasting scientific computing platform
- Linux adoption is growing, as are cluster sizes
- ***Programming*** environments have become mostly stable, but the ***system software*** and management of production systems continues to either:
  - rapidly develop and evolve (glass half-full)
  - grow increasingly fragmented (glass half-empty)
- This tutorial will reveal those cutting edge techniques and processes evolving in the Linux community

Slide 14

Cluster Architecture and Usage

Slide 15

What is a cluster?

- A *cluster* is a collection of interconnected computers used as a unified computing resource. (Pfister)

- Diverse cluster uses
  - High performance compute servers (e.g. scientific computing) < our focus
  - High capacity compute servers (e.g. making movies)
  - Data servers (e.g. Google)
  - High availability/performance databases (e.g. Oracle)

- Depending on use, clusters can be built/optimized for
  - High performance
  - Large capacity
  - High availability
  - Incremental growth

Slide 16

“Beowulf” Clustering

- Clustering of x86-based Linux machines for scientific computing was popularized by the Beowulf project at Caltech/JPL.
- “*Beowulf-class*” usually implies:
  - Off-the-shelf parts
  - Low cost LAN
  - Open source OS
  - Personal supercomputing
- From the Beowulf community we borrow an insistence on
  - Off-the-shelf hardware
  - Linux as the base operating system
- This tutorial departs from the Beowulf approach in
  - Focus on multi-user production (“data center”) use
  - Use of high performance (=expensive) networks when needed by applications.
  - Recommendation to buy commercial software in some cases (e.g. Fortran compiler, PBS Pro, TotalView)

Slide 17

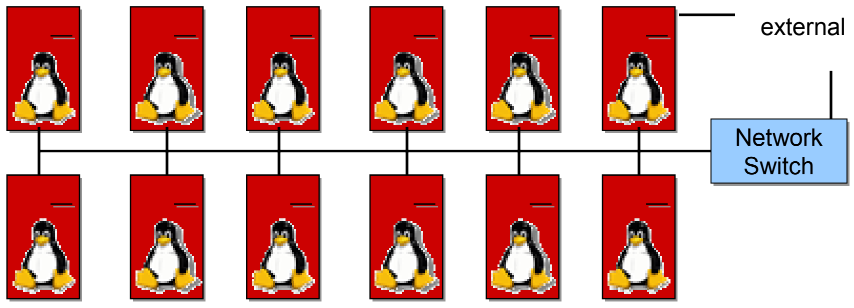
Cluster as a collection of nodes

12-node cluster

- Each node runs a separate copy of the OS (Linux)
- Uniprocessor, SMP, vector, x86, Athlon 64, G5 – can be anything
- 2x performance per processor preferable to 2x processors in many cases
- Uniform nodes are good (more on this later)

Slide 18

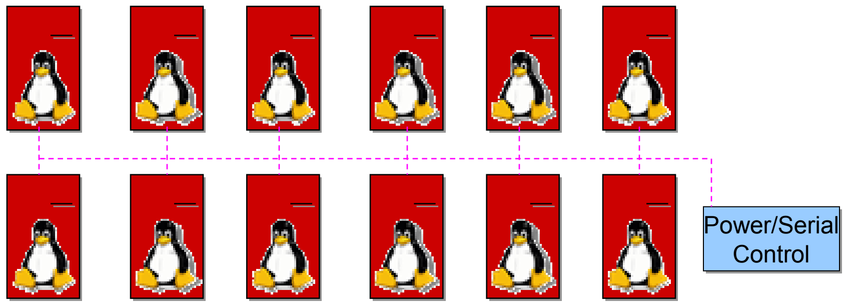
Control network/Data network



- Control network used for management tools, interactive logins, most TCP traffic
- Data network used for interprocess communication in MPI programs
- May be two physical networks, or combined. Two networks gives you reliability and better application performance.
- External network connection can be to the switch or to a node (very different models)
- Much more on networks later

Slide 19

Power/Serial Console



- Must have the ability to power cycle nodes remotely
- Must have ability to see boot messages and log in through the console
  - Serial console preferable to KVM because of cabling/cost/density issues

Slide 20

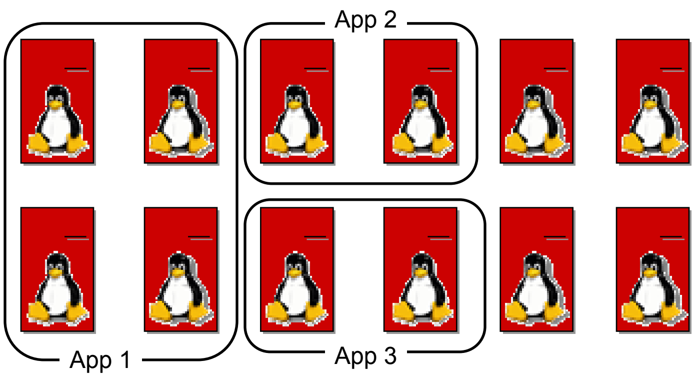
Parallel Application Scheduling

Aside on tightly coupled parallel applications

- Written using MPI
- Network latency/bandwidth matter
- Tight synchronization means 100% dedicated processors are critical
  - 10 µs latency is useless if you have to wait 10 ms (1000x longer) for the receiving process to be scheduled or if the sender is delayed by 10 ms
  - Running two parallel apps on the same set of nodes can result in each app getting much less than half the performance
- Cluster requirements
  - Nodes must be dedicated, not load balanced
  - One MPI process per processor
  - No non-essential system processes on nodes running MPI apps
  - Beware cluster software that uses load average for scheduling

Slide 21

Space Sharing



- Parallel applications must be "space-shared" instead of "time-shared"
  - Nodes are partitioned among parallel applications
  - Scheduling algorithms for space sharing can become quite complex
  - Users do not directly control which nodes their application runs on

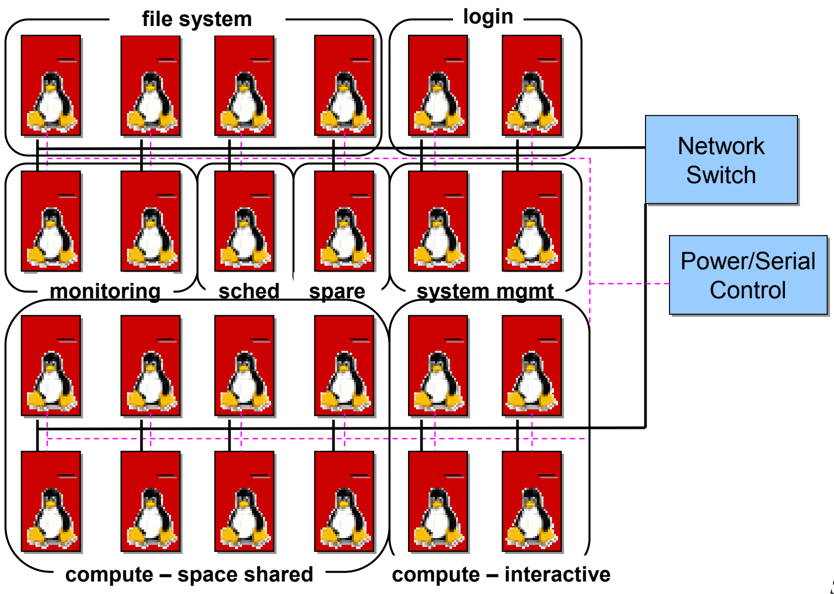
Slide 22

Node Differentiation

- Clusters require more than compute nodes. Usually we dedicate nodes or sets of nodes to other roles
  - Login/compile nodes
  - Space-shared compute nodes for parallel jobs
  - Time/space-shared compute nodes for testing/debugging (scheduling is easier)
  - System management (software distribution/log collection/admin tasks/etc)
  - Monitoring
  - Filesystems
- The larger the system, the more of these you will want to put on dedicated resources. Simplest system has one master/frontend, with the rest as compute nodes
- Some functions may need multiple nodes for performance and/or redundancy.

Slide 23

A Full Cluster



Slide 24

A Bit of Cluster Design Philosophy

- Clusters are typically "buy once". Adding incrementally is often harder than just buying another.
- If at all possible, cluster compute node hardware should be completely uniform.
  - Likewise, all the servers should be as uniform as possible to give you flexibility with swapping parts around.
- Design/plan for scalability. 2x cluster size should not be 2x work.
- Design for recoverability and/or fault tolerance.
  - The more hardware you have, the more things break.

Slide 25

Overview: Early Decisions

Slide 26

Early Decisions

Five major decisions set the context for all of the other decisions on the cluster:

1. The usage model.
   - What is this cluster for? Who will use it, and how will they use it? What are the success criteria?
2. The node architecture.
   - What are the characteristics of the individual nodes?
3. The network architecture.
   - How is the system interconnected?
4. The storage and I/O architecture.
   - How do users access and store their data?
5. The software model.
   - What software runs on the system and how is it managed?

Having covered the usage model as part of the background, we will discuss the node architecture, then network, and so on down the list.

Slide 27

Node Architecture

Slide 28

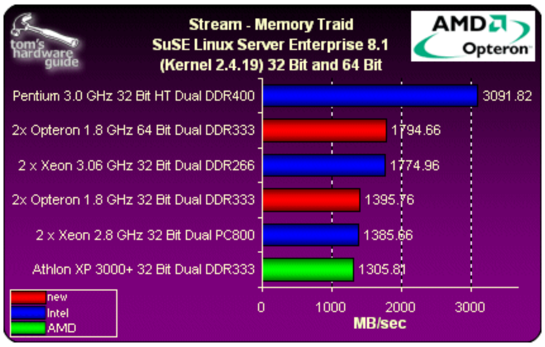
Basic Choices and Components

- CPU
  - x86-32 (Pentium-class)
  - x86-64 (AMD Athlon/Opteron)
  - IA64 (Intel Itanium)
  - PPC/G5
- Uniproc or SMP
- Networking
  - TCP/IP
  - Fast interconnect:
    - Myrinet, Quadrics, Infiniband, GigE
- Rack Form factor
  - 1U, 2U, Blade
- Power Management
  - No management
  - Net-addressable controllers
  - Embedded service processors
- Basic Requirements
  - Space / Cooling / Power
- Console management:
  - BIOS/IPMI/NVRAM
  - VGA
  - Serial

Slide 29

Bandwidth: the “Super” in Supercomputing

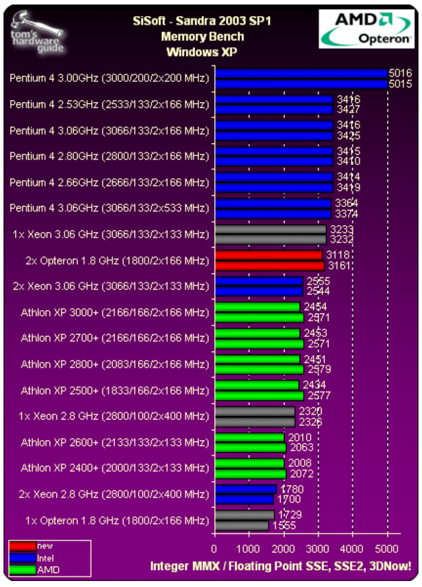
- **Bandwidth costs money**
- CPU/Memory bandwidth is the limiting factor for many applications
- Know your applications
- Peak Theoretic Flops can be deceiving

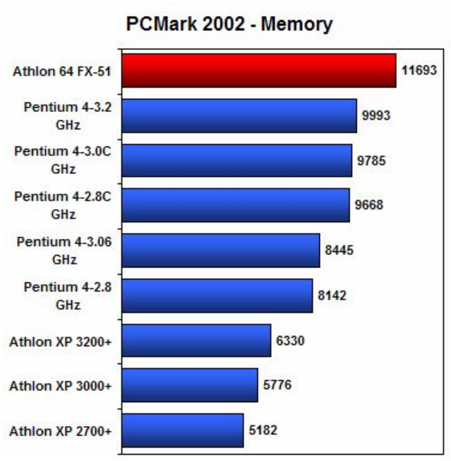


- Perfect World:
  - A[x] = B[i] * C[j] + d; requires 24 bytes/flop memory bandwidth
- Very useful calculation: Aggregate memory BW / $
  - Uniproc vs SMP?
- Benchmarks: See Stream Benchmark, Tom’s Hardware, Sharky Extreme

Slide 30

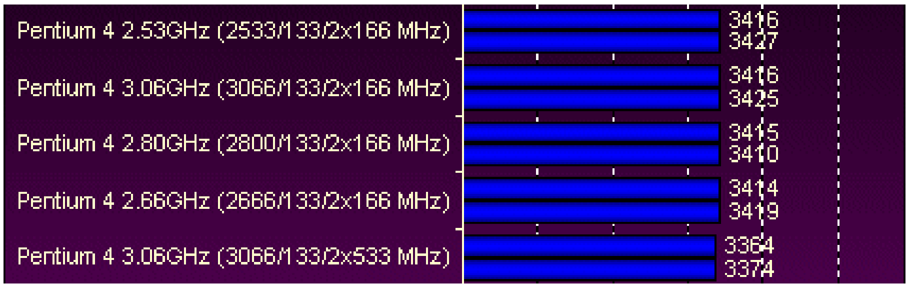
Good Data is Available





Slide 31

Look Closer….



- Notice how the memory subsystem performance remains flat across all these models?
- The P4-3.06Ghz = $456
- The P4-2.53Ghz = $243

Slide 32

Compute Nodes: Other Considerations

- The motherboard implementation can be very important:
  - PCI compatibility issues
  - Dual 64-bit PCI (disk & interconnect)
  - SMP memory performance
  - Memory controller chipset, ECC, speed
  - BIOS / NVRAM / IPMI / Power
- Integrated components:
  - 100BT, VGA, Disk
- Node style issues:
  - Diskless (net boot, NFS root)
    - Advantages: simplified? management, reduced cost, improved reliability
    - Disadvantages: requires scalable netboot, sophisticated software
  - No floppy, cdrom (netboot initial image)
  - No VGA (serial console only)
  - Blades

Slide 33

SMP or Uniprocessor Nodes

Disadvantages to SMP:

- Complicates many layers of software
- Intra-SMP node message passing degrades memory bandwidth
- Overall memory bandwidth per CPU is usually reduced
- Memory can be more expensive (high density usually required)

Advantages to SMP:

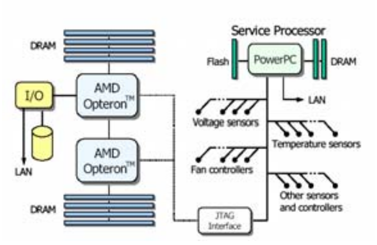
- Cost of fast interconnection fabric split over 2 CPUs
- Compact form
- Price/Performance can be better than uniprocessor nodes

Slide 34

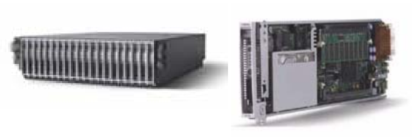
Managed Nodes & Blades

- Many "server" vendors include an embedded "service processor" for "lights out" remote management
- Service processor per node or per enclosure (blades)
- Virtual devices (floppy, CDROM)
- Sensors / controllers
  - RAS features
- Access to BIOS settings remotely
- Out-of-band control
- Hot-plug blades

- Unfortunately, while cool hardware is being developed, the software environments to manage entire clusters is still poor



Example from NEWISYS.com



ProLiant Blades from HP

Slide 35

Control Fabric (1)

- Power Management Options:
  - None. Physically walk to the machine room when you need to cycle a node.
    - Save money, burn more calories, live close….
  - Use a standard, commercial net-accessible AC controller
    - BayTech
    - APC
    - Cost: ~$40/port
    - Net access often uses raw telnet, requiring a protected network
  - Compute nodes must have right "power button"
    - Bad: momentary contact button must be physically touched
    - Does On/Off switch retain state?
  - Software utilities – still poor
  - IPMI/EMP – http commands over tcp/ip, messy to set up
  - Large hardware integrators provide easy-to-use tools (IBM/LNXI/etc)

Slide 36

Control Fabric (2)

- Linux Console Management:
  - Shopping cart, bungee cord, monitor, and VGA cable
  - Very small clusters can use VGA switch
  - Serial console:
    - Physically connect all the serial ports into the control node via multi-port serial card
    - Comtrol's Rocketport:
    - Cyclades:
    - Requires software infrastructure: conswatcher, chex, conserver
  - Emergency Management Port (EMP)
    - BIOS access, power/reset, NVRAM critical event log, sensor data
  - Large hardware integrators offer reasonable solutions

Slide 37

The BIOS

- The BIOS controls what the machine does when it powers on.
- The average PC BIOS is harder to work with than the average UNIX workstation BIOS:
  - No command line.
  - No remote access.
  - No OS-level access, at least not for Linux. Yet.
- BIOS configuration settings are set at the factory.
  - You might be happy with what you get, you might not.
- Options for fixing the BIOS:
  - Plug in a monitor and keyboard.
  - Some BIOS have serial port access and upgrade features, they work, but messy.
  - LinuxBios, motherboard specific, but many common ones available now.
  - You might be able to use the software tool bioswriter – won't work with all.
  - Boot into DOS (this is possible with netboot), run flash tool, can automate, but still an art.

Slide 38

Evaluation/Design Checklist

- Node performance
  - Suggested CPUs: Opteron, Intel Pentium 4, Athlon FX-51
  - CPU/Memory bandwidth is key
- Node manageability
  - Compatibility: is your configuration a common one?
  - Lights-out functions for BIOS, status, sensors, etc
- Control fabric
  - Lights-out for power, consoles, etc
- RAS (ask about it!)
  - Current Linux clusters take very little advantage of RAS features in hardware, little if any real Open Source software exists in this domain (ECC is one example)

Slide 39

Schedule: Morning, Part II

Slide 40

Morning, Part II

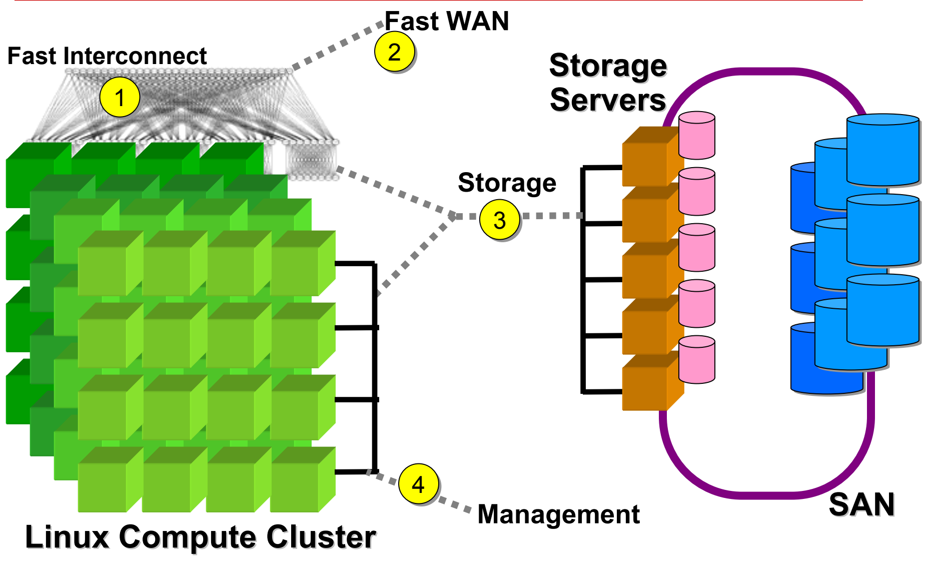
- Network Architecture Options – Pete and Bill
  - Network performance is fundamental to the overall performance of the cluster, and can also be extremely costly.
- Storage and I/O Options – Remy and Susan
  - Large-scale storage for clusters is still difficult, but users expect it to just work. Here are the options.
- Purchasing and Hardware Installation – Susan
  - The issues to be aware of when purchasing and installing your cluster.

Slide 41

Network Architecture

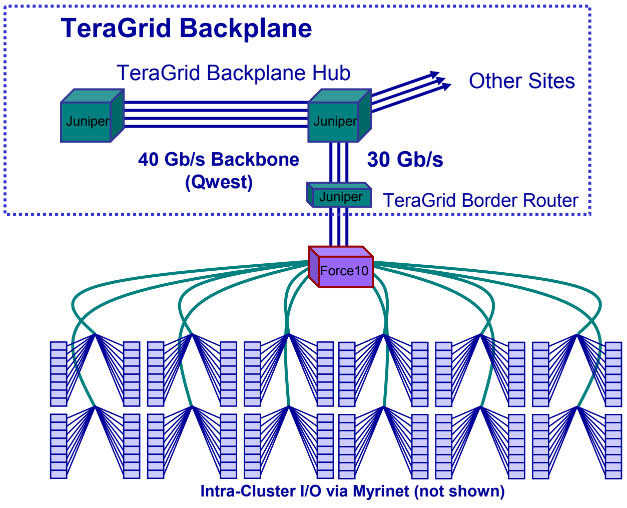
Slide 42

Basic Networking Uses

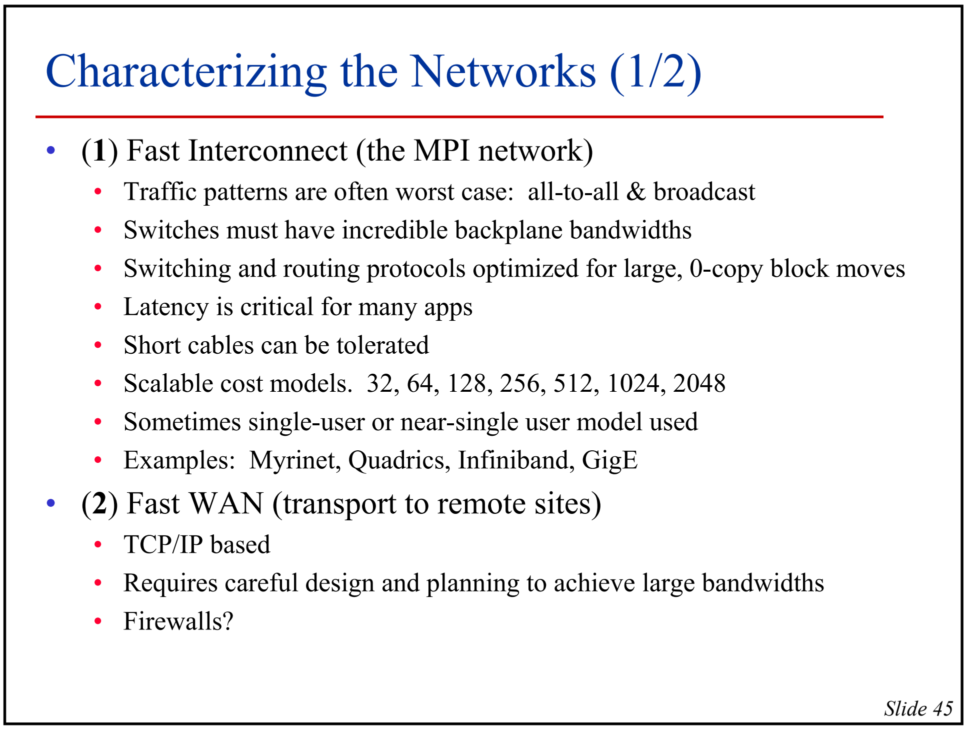


Slide 43

TeraGrid Example



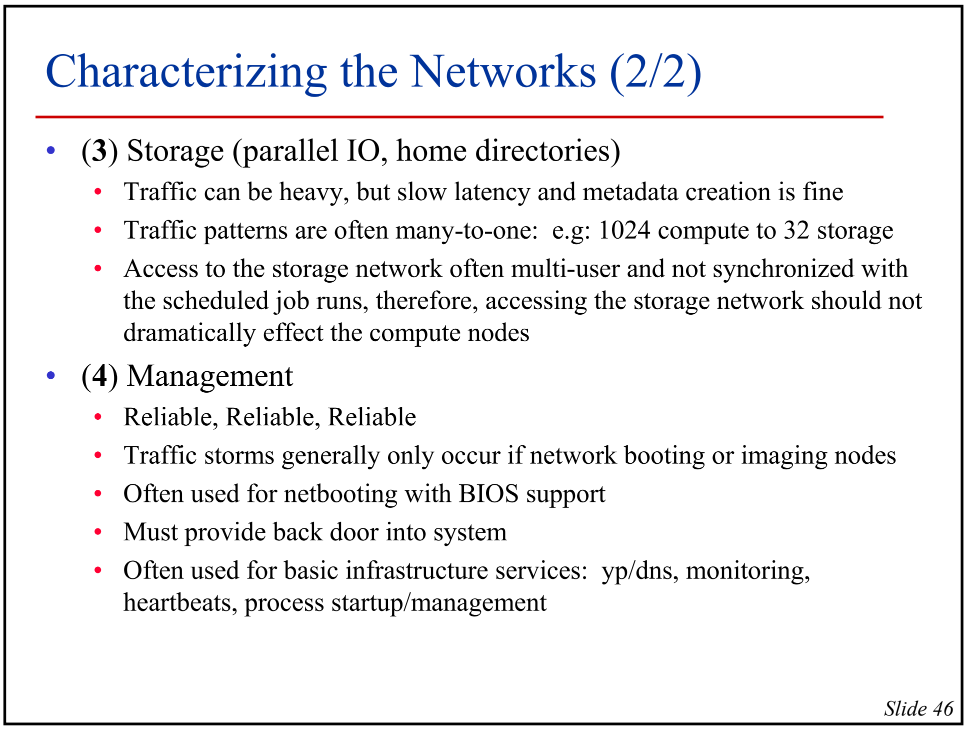
Slide 44



Characterizing the Networks (1/2)

- (**1**) Fast Interconnect (the MPI network)
  - Traffic patterns are often worst case: all-to-all & broadcast
  - Switches must have incredible backplane bandwidths
  - Switching and routing protocols optimized for large, 0-copy block moves
  - Latency is critical for many apps
  - Short cables can be tolerated
  - Scalable cost models. 32, 64, 128, 256, 512, 1024, 2048
  - Sometimes single-user or near-single user model used
  - Examples: Myrinet, Quadrics, Infiniband, GigE
- (**2**) Fast WAN (transport to remote sites)
  - TCP/IP based
  - Requires careful design and planning to achieve large bandwidths
  - Firewalls?

Slide 45



Characterizing the Networks (2/2)

- (**3**) Storage (parallel IO, home directories)
  - Traffic can be heavy, but slow latency and metadata creation is fine
  - Traffic patterns are often many-to-one: e.g: 1024 compute to 32 storage
  - Access to the storage network often multi-user and not synchronized with the scheduled job runs, therefore, accessing the storage network should not dramatically effect the compute nodes
- (**4**) Management
  - Reliable, Reliable, Reliable
  - Traffic storms generally only occur if network booting or imaging nodes
  - Often used for netbooting with BIOS support
  - Must provide back door into system
  - Often used for basic infrastructure services: yp/dns, monitoring, heartbeats, process startup/management

Slide 46

(remember this?)

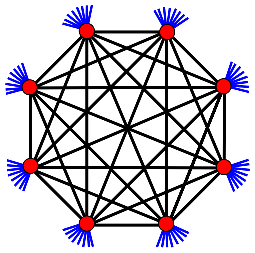
Bandwidth: the "Super" in Supercomputing

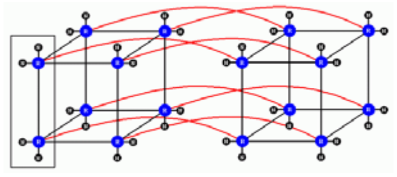
- **Bandwidth costs money**
  - Typically about 30% of system cost is the Fast Interconnect
- ***Fast Interconnect*** bandwidth can be the limiting factor for many applications
- Know your applications
- Peak theoretic link speeds can be deceiving
- The tao of balance: link speed, PCI bus speed, NIC $, switch $
  - A top-of-the-line Opteron server can achieve about 985MB/s over the PCI-X bus to memory with Myrinet card.
  - Can you use the link bandwidth you purchased?
  - Can your PCI-X bus use the Fast Interconnect bandwidth you purchased?
- Do the math, do the benchmarks, find the sweet spot
- Examine bytes/flop balance

Slide 47

Definition: Bisection Bandwidth

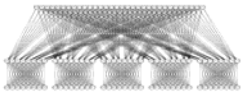
- Bisection Bandwidth is used to report the connectivity of the entire cluster, in contrast to "latency" and "bandwidth" which are measured only between a pair of hosts.
- Theoretic Bisection Bandwidth is calculated by dividing the machine in half, using the partition of worst connectivity, and summing the bandwidth of the links between the halves.
- Interconnection topologies have "Full Bisection Bandwidth" when the number of links between any two halves of the machine is N/2.
- Bisection Bandwidth is important for programs that do global communication





Slide 48

Fast Interconnect Topology



- Today, most clusters use interconnect topologies with generous amounts of bisection bandwidth (Clos network, fat tree, etc)
- This implies:
  - The scheduler does not need to map nodes together onto the topology for performance – any nodes can run together
  - The interconnection network switches are usually not the bottleneck
  - Irregular communication patterns are easily supported
  - ***Interconnect cost is high, and harder to scale***
- Alternative: closer-to-linear scaling, for example 3D-Torus
  - Best performance achieved with scheduler assigning "slices" and "bricks"
  - Irregular communication patterns likely to cause performance issues
  - Interconnect cost grows more linearly with node count
  - Used by Cray X1, Sandia Red Storm, IBM Blue Gene for very large systems

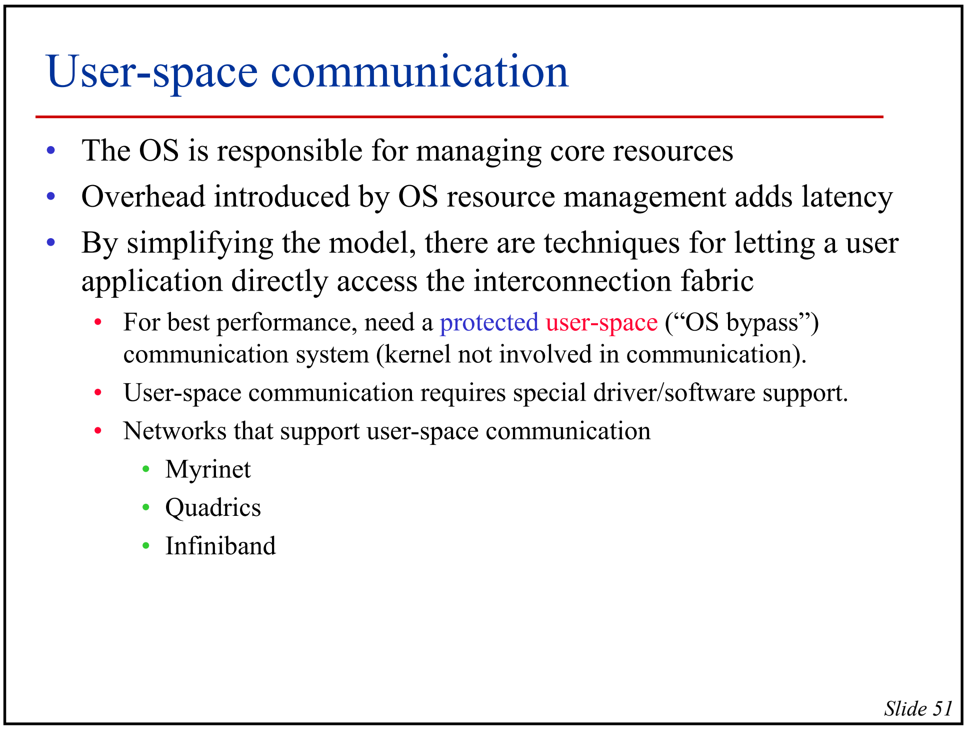
Slide 49

Network Performance

(The numbers you see may not be the numbers that matter)

- Real performance is a combination of several important factors
  - NIC, media, duplex, switches, contention, hot-spots, etc.
  - Software (TCP/IP Stack, user-level messages, etc)
  - CPU Utilization (benchmarks can be ***very*** misleading here)
    - The MPI latency lie…
    - Zero-copy tag matching
  - System parallelization, asynchronous data movement
- There are two important network usage patterns for clusters
  - Scientific message passing for parallel programs (MPI)
    - Users want:
      - 1) low latency, high bandwidth, no network hotspots
      - 2) Asynchronous data movement, low CPU cost
  - SAN/WAN (TCP/IP)
    - Fast disk, tape, grid, home directory NetApp access

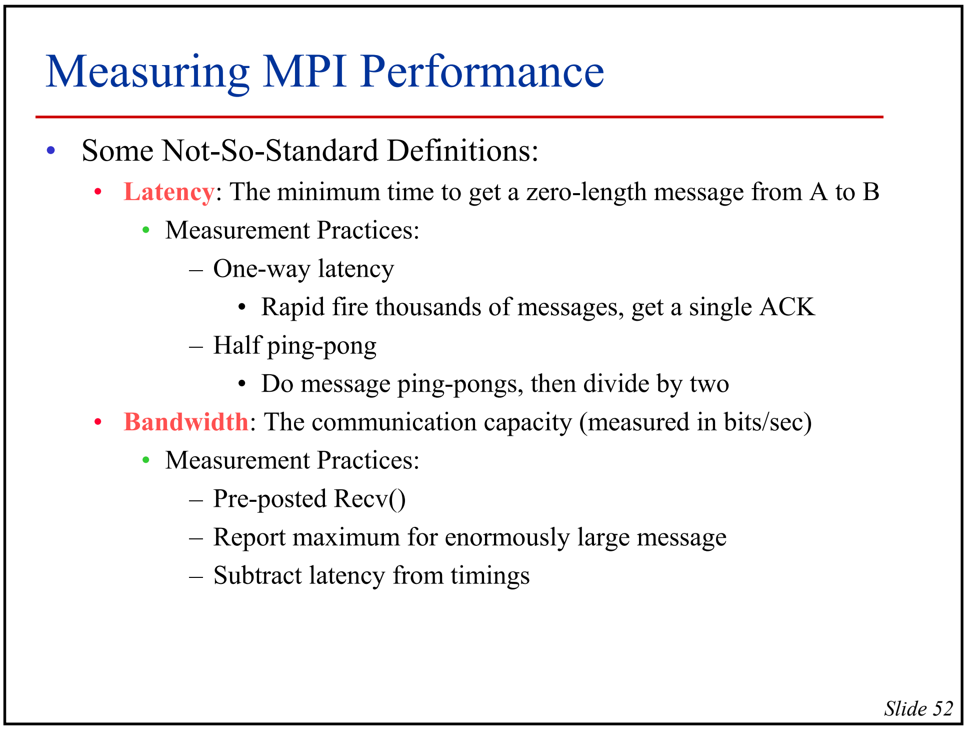
Slide 50



User-space communication

- The OS is responsible for managing core resources
- Overhead introduced by OS resource management adds latency
- By simplifying the model, there are techniques for letting a user application directly access the interconnection fabric
  - For best performance, need a protected user-space (“OS bypass”) communication system (kernel not involved in communication).
  - User-space communication requires special driver/software support.
  - Networks that support user-space communication
    - Myrinet
    - Quadrics
    - Infiniband

Slide 51



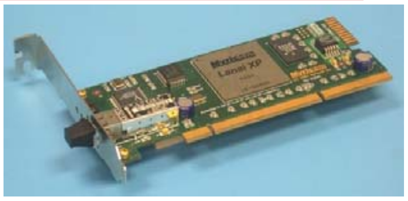
Measuring MPI Performance

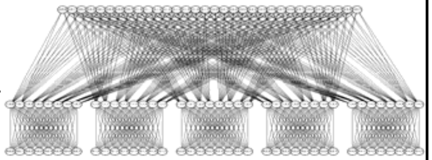
- Some Not-So-Standard Definitions:
  - **Latency**: The minimum time to get a zero-length message from A to B
    - Measurement Practices:
      - One-way latency
        - Rapid fire thousands of messages, get a single ACK
      - Half ping-pong
        - Do message ping-pongs, then divide by two
  - **Bandwidth**: The communication capacity (measured in bits/sec)
    - Measurement Practices:
      - Pre-posted Recv()
      - Report maximum for enormously large message
      - Subtract latency from timings

Slide 52

Myrinet

- Basic product line
  - 133 Mhz PCI-X NICs
  - 985 MB/s PCI to memory, best case
  - 250 MB/s + 250 MB/s
    (bidirectional) links (fiber)
  - Open Source*, user-level msg drivers for MPICH
  - High-bandwidth interconnect topology constructed from 16-port switch chips
- Recent addition: Blade form factor

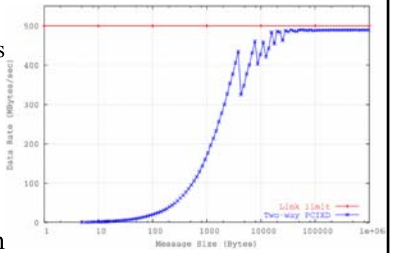




Slide 53

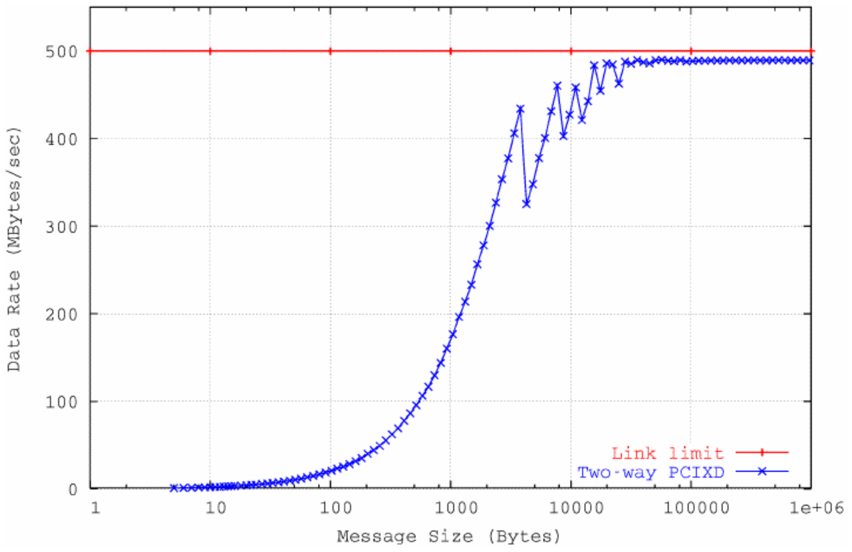
Myrinet: More Details

- Features
  - Flow control, heartbeat monitoring; GigE cards available for interoperability
  - Very mature product line
- 178 of the June 2003 Top500 use Myrinet (highest is #11; many are resold by HP)
- Supported by both MPICH and LAM/MPI
- Low host CPU overhead (MPI tag matching in NIC) (.8us for short msg)
- Short message latency: 6.3 µs (best configs)
- Sustained two-way, large messages: 489 MByte/s
- Moderately expensive (~$1500/node)
- Relatively good tao of balance (for Linux):
  - Delivered msg-passing performance about ½ max MoBo performance for single-port NIC



Slide 54

Myrinet Performance



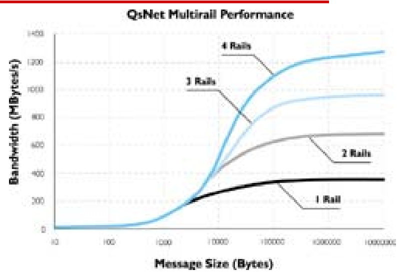
Slide 55

Quadrics

- Basic product line (QsNet)
  - 64 bit, 66MHz PCI Bus
  - 528 MB/s PCI to memory, best case
  - 340 MB/s + 340 MB/s

    (bidirectional) links (copper, YUK!)
  - Open Source drivers, hooks to virtual memory system significant, user-level msg supported
  - High-bandwidth interconnect topology constructed from 8-port switch chips
  - Unique mapped virtual memory for SHMEM put/get
  - Supports multiple “Rails” (especially for large SMPs)

  - QsNet II will be better, faster, etc…



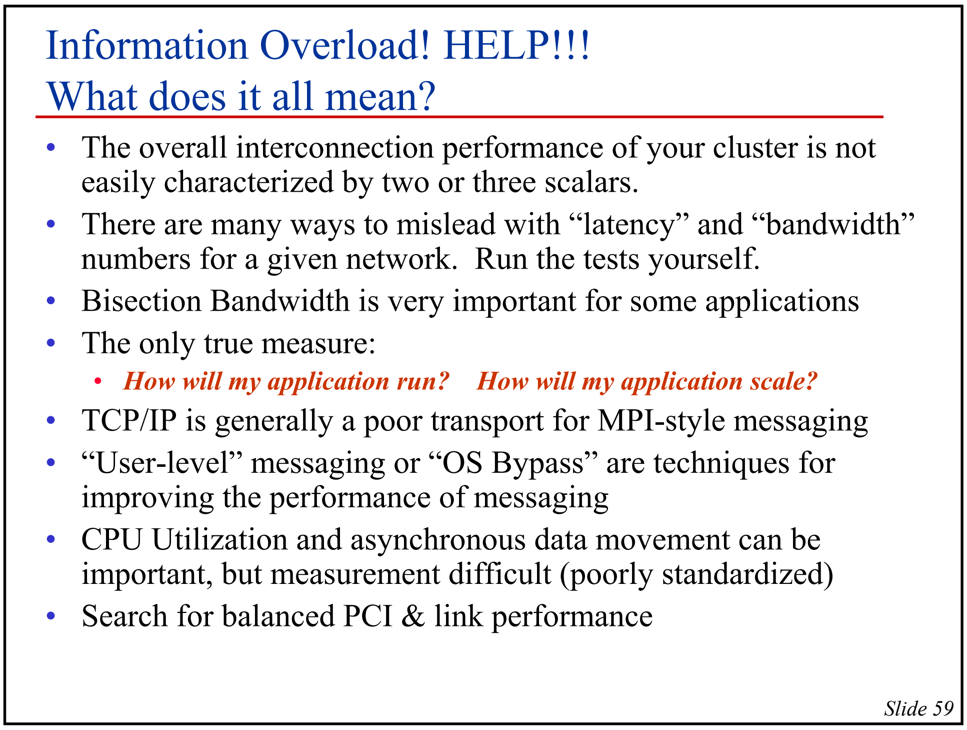
Slide 56

Quadrics: More Details

- Features
  - Elan 3 cables are terrible. Elan 4 rumored to be better.
  - ***Key feature: very low-latency put/get***
- MPI Performance: 4.6 us latency; >300 MB/s
- Poor CPU overhead: 3.3us for short msg
- Requires the use of Quadrics software (RMS and MPI). There is some PBS integration. Nonstandard software has large learning curve and stability is an issue.
- 27 of June 2003 Top 500, including #2,3,6,8,9,10
- Very expensive (~$3k/node???)
  - You know it is expensive if the prices are not listed on the web, and the web site says "contact a salesman". Eeew!
- QsNet I has ok tao of balance (for Linux)
  - Delivered msg-passing performance about 1/3 max MoBo performance for single-port NIC

Infiniband

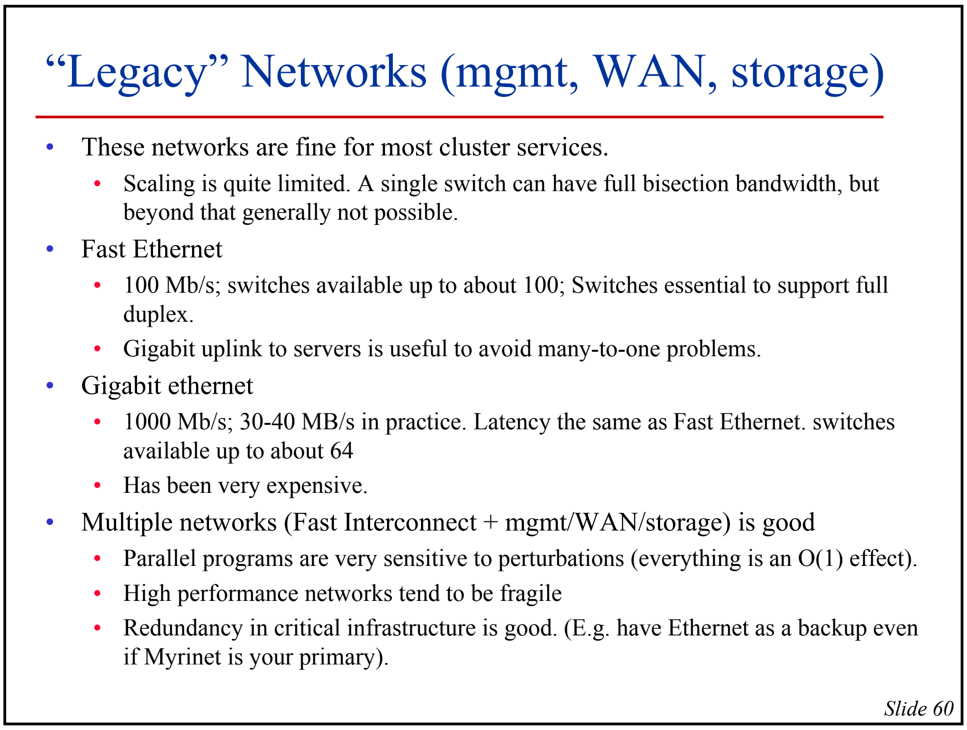
- Features
  - A "standard" that has been a long time coming. Recently re-emerged as a cluster interconnect when it failed as an internal system bus
  - Switched network; interoperable; 10 Gb/s peak performance.
  - Almost no real-world data, but several large clusters are being built with Infiniband – keep an eye on it.
  - No big switches
- MPI Performance: 6.8 us latency; 700 MB/s bandwidth
- Multiple vendors. E.g.
  - Mellanox Technologies
  - Topspin Communications
  - InfiniCon Systems
  - Existence of many hardware vendors may make Infiniband a commodity inexpensive product. Right now it is "bleeding edge"
- Software for Infiniband products is not yet mature



Information Overload! HELP!!! What does it all mean?

- The overall interconnection performance of your cluster is not easily characterized by two or three scalars.
- There are many ways to mislead with "latency" and "bandwidth" numbers for a given network. Run the tests yourself.
- Bisection Bandwidth is very important for some applications
- The only true measure:
  - ***How will my application run?***   ***How will my application scale?***
- TCP/IP is generally a poor transport for MPI-style messaging
- "User-level" messaging or "OS Bypass" are techniques for improving the performance of messaging
- CPU Utilization and asynchronous data movement can be important, but measurement difficult (poorly standardized)
- Search for balanced PCI & link performance

Slide 59



"Legacy" Networks (mgmt, WAN, storage)

- These networks are fine for most cluster services.
  - Scaling is quite limited. A single switch can have full bisection bandwidth, but beyond that generally not possible.
- Fast Ethernet
  - 100 Mb/s; switches available up to about 100; Switches essential to support full duplex.
  - Gigabit uplink to servers is useful to avoid many-to-one problems.
- Gigabit ethernet
  - 1000 Mb/s; 30-40 MB/s in practice. Latency the same as Fast Ethernet. switches available up to about 64
  - Has been very expensive.
- Multiple networks (Fast Interconnect + mgmt/WAN/storage) is good
  - Parallel programs are very sensitive to perturbations (everything is an O(1) effect).
  - High performance networks tend to be fragile
  - Redundancy in critical infrastructure is good. (E.g. have Ethernet as a backup even if Myrinet is your primary).

Slide 60

Network Configuration Issues

- Public vs. Private network addresses
- Routing via switch or host (including masquerading)
- How to configure multiple networks

Slide 61

Private Networks

- Three sets of network addresses are reserved for anyone’s use. Just don’t let packets with these addresses out of your cluster!
  - 10.0.0.0 - 10.255.255.255
  - 172.16.0.0 - 172.31.255.255
  - 192.168.0.0 - 192.168.255.255
- Security. Complete trust inside a private network is possible.
  - Firewalling/packet filtering less complicated.
  - Some tools rely on complete (rsh-style) trust.
  - Many services do not need to be exposed outside the cluster
- Allocation of IP addresses is easier. Automatic assignment to nodes is easier.
- Simplifies novice setup: no interference with existing DHCP, NFS, YP or other services
- No interference from traffic external to the cluster.

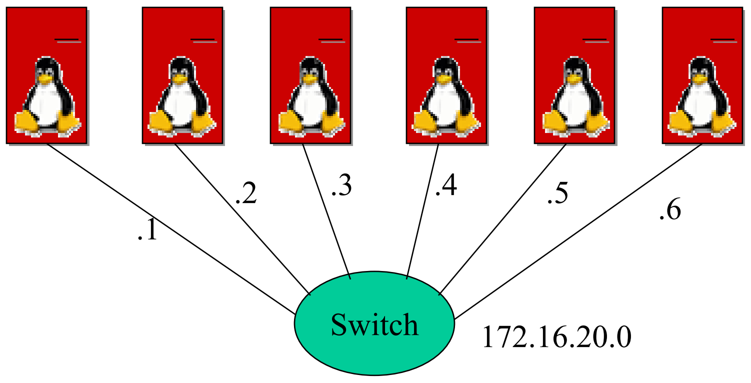
Slide 62

But…

- Generally requires dual-ported manager node to provide all basic cluster infrastructure services:
  - DNS, DHCP, NFS, YP, etc
  - Hardware management & monitoring
- Some users want connections to the outside world directly from cluster compute nodes.
  - MPI jobs that span multiple clusters
  - Grid-based applications that make direct connections from compute nodes to central service
  - Application “portals”
  - Parallel IO servers may need to be on public network
- Complicates management when those services already exist and are supported by department.

Slide 63

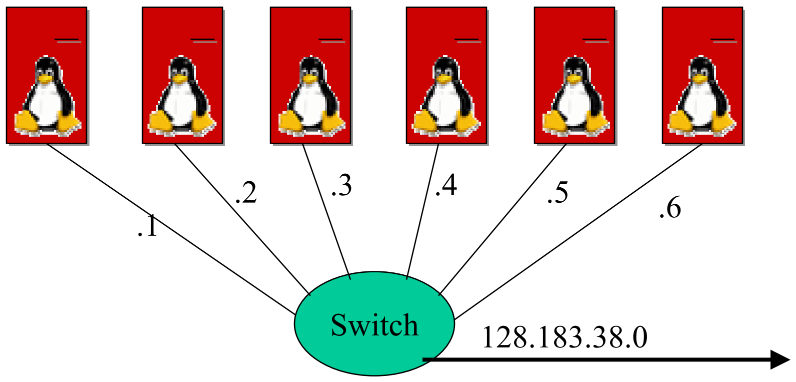
A Private Network



Slide 64

A Public Network

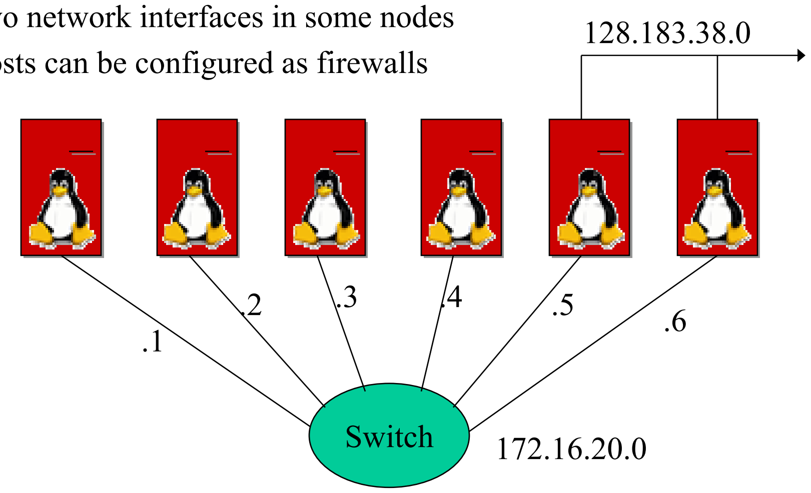
- All nodes are "on the internet"



Slide 65

Gateway Nodes (+ optional Masquerading)

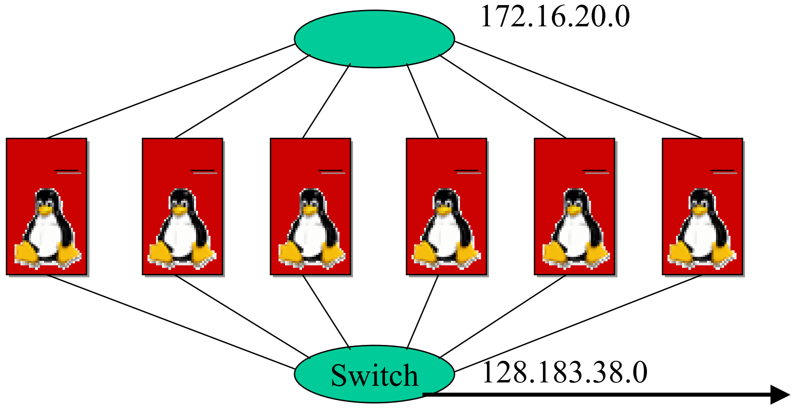
- two network interfaces in some nodes
- hosts can be configured as firewalls



Slide 66

Two networks: 1 public, 1 private

- Routing/firewalling can be complex



Slide 67

Summary of Functionality

- Fast Interconnect: MPI
  - node-to-node communication
  - requirements: low latency, high bandwidth, scalability. Not TCP
- Storage
  - compute-node-to-storage-node communication
  - requirements: high bandwidth, scalability
  - Often the same as the MPI network
- Management
  - software distribution, monitoring, access to power/console, interactive logins.
  - must support TCP
- External connectivity
  - connects cluster to the outside world
- Storage area network (SAN)
  - storage-node-to-disk communication

Slide 68

Network Summary

- For tightly coupled parallel applications
  - Get a high performance switched network with OS bypass
  - Expect 30% of cluster cost to be network
  - Install a second network (fast or gigabit Ethernet) for control/management
- High performance network choices
  - Myrinet – fast enough for many applications; very common
  - Quadrics – maybe faster for some functions, but very expensive and difficult to manage
  - Infiniband – not yet proven. Keep a close eye on it.
- Public vs. private network
  - Depends on whether application compute nodes need to talk to outside world, and how you transfer data to/from cluster storage
  - Administration may be easier for novices, but very difficult in centrally managed environments that already support many services

Slide 69

Storage and I/O Options

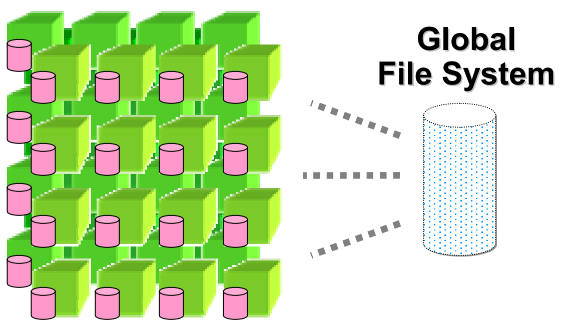
Slide 70

Storage for Clusters is Difficult

- Users want (and need):
  - Everything
- This is a puzzle with many pieces that may not quite fit your needs.
  - There are many technologies, projects, and companies.
  - Understanding just what a particular technology solves can be difficult.
  - The solutions overlap but don't compare cleanly.
  - Scaling to large systems remains a difficult issue for all solutions.
- There is not, yet, a simple solution for Linux cluster storage.

Slide 71

Filesystem Terminology



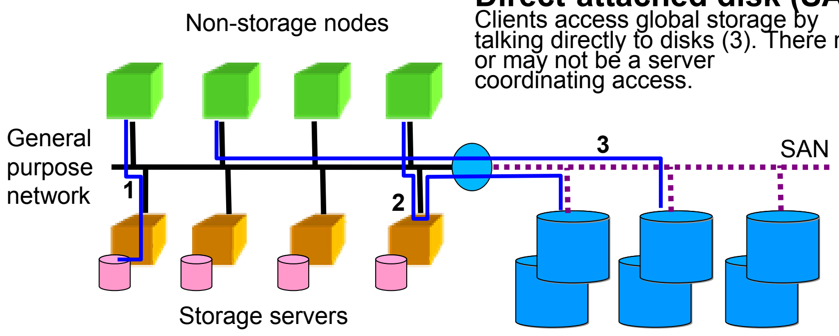
Local Disks/Filesystems
Only accessible on its node.

Global File Systems
Accessible across entire cluster.
Single name space

Parallel File System
A global file system in which performance can increase when you add either servers or clients. Supports cooperative access of several clients to a single file (esp. writing)

Slide 72

Storage Architecture: how clients access data



Direct-attached disk (SAN)
Clients access global storage by talking directly to disks (3). There may or may not be a server coordinating access.

Storage Servers (NAS)
Clients access global storage by connecting to storage servers over a network. Storage servers coordinate access. They may have local disk (1)or be connected to disks by a SAN (2)

Hybrid
Clients talk to a server for metadata but talk to disks directly to read/write data (2+3)

Slide 73

Why Is Cluster Storage Difficult?

- Maintaining consistency and coherency of data and metadata is difficult.
- Semantics of parallel access is hard, esp. cooperative writing
  - The MPI-IO API allows an application to define processes cooperatively access a single file.
  - POSIX semantics require nonoverlapping writes to work (they do not work correctly in NFS)
- Performance is hard
  - Scalable performance requires that different clients magically write to different physical disks (e.g. By striping across disks and using separate channels)
  - Locking implicitly required for metadata coherency can serialize fs access.
- Data sizes vary widely
  - Some applications dump TBs of data during a run in a single file
  - Other applications dump thousands of small files
  - Everyone wants access to their home directory (startup files, etc)
- Direct access to data from other machines is important in many apps.
- Doing all of this at the same time is very difficult.

Slide 74

Storage Options

- Disk technology
  - Individual disks, JBODs, RAIDs, volume managers, storage subsystems (e.g. EMC)
- Host connection
  - Direct attached disk: IDE, SCSI, Firewire
  - SAN-attached disk: Fibre Channel, Infiniband, iSCSI
- Type of file system
  - Local filesystems: Reiser, xfs, jfs, ext3
  - Network file systems: NFS, PVFS, GPFS (IBM), Lustre (there may be an underlying local filesystem), StorageTank (IBM)
  - Multiple host direct-attached filesystems: GFS, ADIC, GPFS (has two modes), StorageTank
  - Hybrid filesystems: DAFS
  - Federated filesystems (presents unified view of smaller filesystems): InfinARRAY, Ibrix
- Other infrastructure
  - Storage nodes (if needed)
  - SAN infrastructure: switches, host adapters
  - IP network infrastructure (may be the “MPI” network)
- Many, many companies sell solutions that cover one or more of these spaces.
  - IBM, ADIC, BlueArc, Panasus, Polyserve, Sistina, …

Technologies useful today or very soon

- Lustre
  - Lots of development and interest. Not yet successfully exported outside of LLNL, but expect it soon.
- PVFS
  - Has been the only free parallel FS option for a while. In use at many places. Performance is decent.
- NFS
  - By far the most common filesystem on small clusters. Ok for small clusters and applications without significant I/O. But very slow for parallel I/O and doesn't support POSIX semantics.
- GFS
  - Robust with decent performance, it can be used directly or with NFS. Costs $.
- GPFS
  - GPFS is the workhorse for the IBM SP. Recently released for Linux, and there is now a serverless (SAN) version. Costs $.

Archiving

- Production computing centers typically have some kind of tape archive library.
  - These are very expensive, but currently the main, viable option for large-scale long-term data archiving.
  - See HPSS , IBM (Tivoli Storage Manager), ADSM.
- Because of the volume of data, this is just hard problem
  - Have a plan.
  - Nothing is fantastic.
  - Choice is all about $$$.
  - Options:
    - Tape archive library – works well, but is very expensive.
    - Hard drive archive – becoming more cost effective, software is still an issue.
    - Don't back up the data files – regenerating the data can be faster and cheaper.

Slide 77

Usage Policies

- Have a storage usage policy in place and published.
  - If you don't have one, then your policy is "any user can fill up the disk".
- Usage enforcement options:
  - Quotas.
  - Monitoring and cleaning up with scripts.
  - Letting users complain about other users.
- Space monitoring
  - Users will use all available space. (Jazz's 10TB lasted about 9 months.)
  - Some filesystems can cause serious problems once they become full.
  - No matter what you choose, make sure that you set up monitoring of any critical filesystems.
- On Jazz, we:
  - monitor the global filesystems and some local disks (i.e. /var/spool/PBS),
  - handle usage on a case-by-case basis.

Slide 78

Our Storage Architecture Advice

- Know what I/O characteristics your users need.
  - If parallel I/O isn't important, don't worry about parallel file systems, etc.
  - Most likely, users will want a global file system (by default).
- Keep in mind that, at present, there is no single ideal solution.
  - Consider multiple file systems, each for different uses.
- Talk to multiple vendors about the solutions they offer. Some have unique solutions, while others fine-tune open source options.

Slide 79

Working Examples

- On smaller clusters (<100 nodes), an open-source option:
  - NFS is typically acceptable at this scale
  - Individual NFS servers for home directories
  - 4-8 servers with PVFS for parallel I/O
- On Jazz, we have:
  - Local disks on each node.
  - A parallel IO scratch space.
    - 8 PVFS servers, each with 1-2 TB of JBOD
    - All users have access for short-term storage
    - Optimized for performance, not reliability.
  - A global file system for home directories.
    - 8 servers, each with 1-2 TB of Fibre Channel RAID
    - GFS running between the servers, creating one shared file system
    - NFS from each server to 1/8th of Jazz
    - Highly reliable, but not high performance.

Slide 80

Overview: Decisions, Purchasing and Installation

Slide 81

Before we discuss the Software decision…

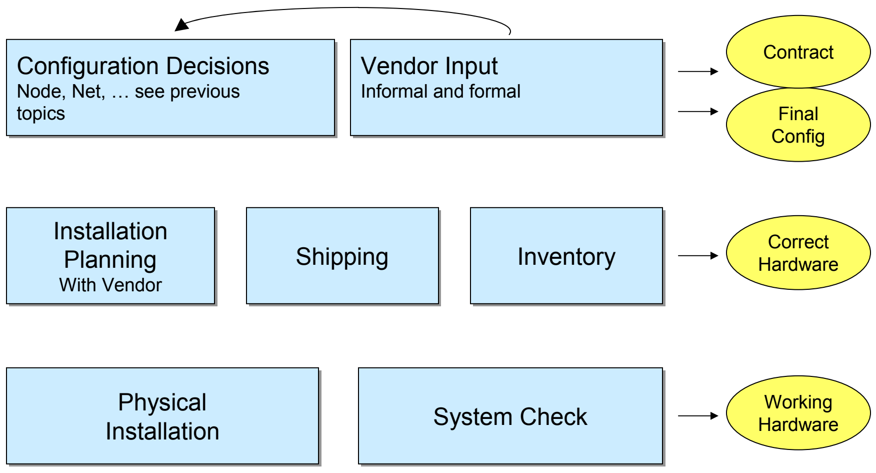
- The 5th major decision you make when planning your cluster is the software model.
- This is a complex and important decision. We will cover more information in this tutorial before we can effectively discuss the software model.
- So, we'll return to the software choice a bit later, after we cover:
  - Purchasing and Installing
  - System Configuration
  - Resource management
  - Applications and Environment

Slide 82

Purchasing and Installation

Slide 83

Purchasing & Hardware Installation – Steps



Slide 84

Configuration Decisions - Step 1
The Constraints

- Your budget
- The goal for the cluster – success criteria
  - Two years after the cluster is installed, what will you need to able to say that the cluster has been used for in order to declare success?
  - The configuration better allow this!
- Physical limitations
  - floor space
  - power and cooling available
- Support staff
  - You will need people to run the cluster and help the users.
  - The more complicated the configuration, the more people.

All of these should be considered even before looking at the technical options.

Slide 85

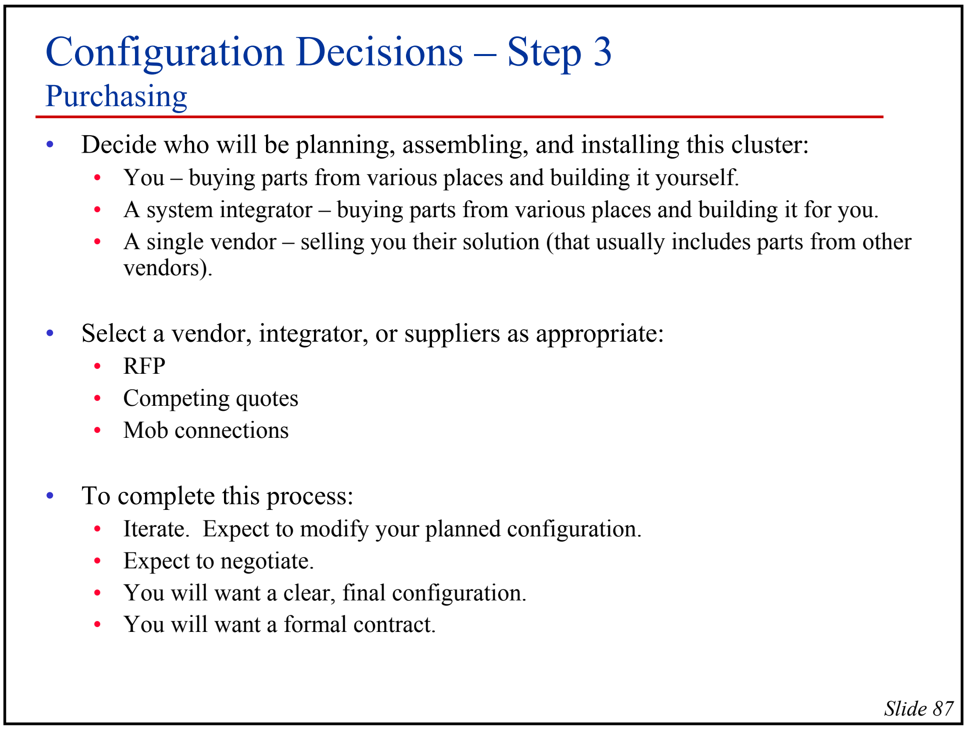
Configuration Decisions – Step 2
Technical Decisions

As we've already discussed in some detail:

- Usage model
- Node architecture
- Network architecture
- Storage and I/O architecture
- Software model

Usually, you won't have a specific configuration in mind, but rather a range of options. The next step is to narrow these down based on cost, availability, functionality, and so on…

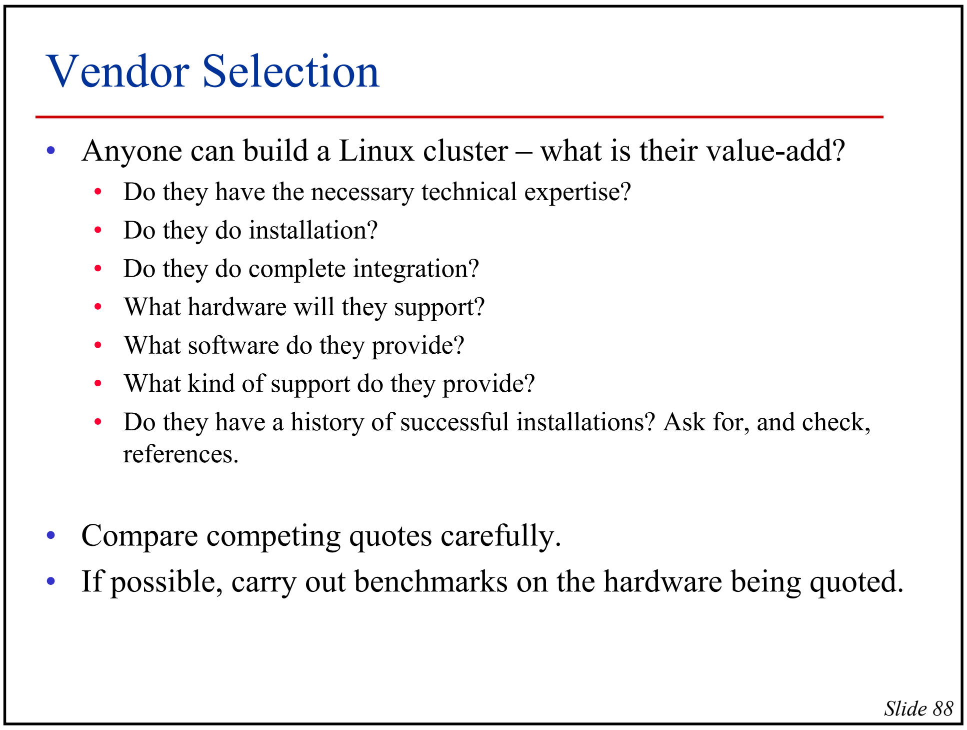
Slide 86



Configuration Decisions – Step 3
Purchasing

- Decide who will be planning, assembling, and installing this cluster:
  - You – buying parts from various places and building it yourself.
  - A system integrator – buying parts from various places and building it for you.
  - A single vendor – selling you their solution (that usually includes parts from other vendors).
- Select a vendor, integrator, or suppliers as appropriate:
  - RFP
  - Competing quotes
  - Mob connections
- To complete this process:
  - Iterate. Expect to modify your planned configuration.
  - Expect to negotiate.
  - You will want a clear, final configuration.
  - You will want a formal contract.

Slide 87



Vendor Selection

- Anyone can build a Linux cluster – what is their value-add?
  - Do they have the necessary technical expertise?
  - Do they do installation?
  - Do they do complete integration?
  - What hardware will they support?
  - What software do they provide?
  - What kind of support do they provide?
  - Do they have a history of successful installations? Ask for, and check, references.
- Compare competing quotes carefully.
- If possible, carry out benchmarks on the hardware being quoted.

Slide 88

Vendor Interaction – Lessons Learned

You will likely see several quotes in the process of settling on a configuration.

- The first quote almost always has some major mistakes.
  - Check the network and storage configurations in detail.
- Pay close attention to the details of **every** quote.
- Keep your own copy of your desired configuration, and refer to that with each iteration.
  - Hint: draw a picture of the cluster on your white board. Include details such as number of systems, type of hardware, number of cables. Compare every quote to this picture.

Be detailed and document everything.

Everything that isn't explicitly stated is a potential future land mine!

Slide 89

Vendor Interaction – Lessons Learned, 2

Important details:

- Who does what and when?
  - Make sure that you and the vendor both know and agree.
- Who pays for the return shipment of the unused or replaced hardware?
- Put in the **manufacturer** part number for the components to avoid confusion.
- What happens if it is late? Or if some piece can't be delivered? Have written 'automatic' rules.
- What *exactly* does 'acceptance' mean? (Here you need to be realistic, careful, and very explicit – see next slide.)
- How will disputes be resolved?
- An integrator or vendor should coordinate all of the various warranties. Get a single point of contact for all of the hardware and software support.
- Make sure you know what the warranty and support model is for everything.
- Discuss backup plans for risky decisions.

Slide 90

Vendor Interaction – The Acceptance Test

- Acceptance tests are important, particularly for large $$ systems.
- Acceptance tests should be clearly documented. The plan should include:
  - A list of exact benchmarks that will be run, with required results.
  - A description of how the tests will be conducted.
  - Definitions of all terms, including the software environment, what "97%" success means,
  - A schedule.
  - A plan for addressing failures.
  - Other criteria, such as cables labeled, software installed, etc
- A few tests to consider:
  - Configuration – hardware and software is installed, functions correctly, passes diagnostics
  - Cluster validation – a series of tests verifying that base cluster functions run correctly
  - Continuous availability – running a long job during which n% of system must stay up
  - System tests – stressing and testing functionality of various system services
  - Benchmark tests – a set of benchmarks (compute, i/o, network, memory, etc) that must be met
  - Customer tests – using your own code to test results and performance
- Typically the sale is complete after acceptance.

Slide 91

Arrival

- Having settled on a final quote and survived the contract negotiation, your goal now is to get the right hardware safely to your site, ready for installation.
- You will be working with the vendor (if you have one) during this time to plan for delivery and installation:
  - Installation Planning
  - Shipping
  - Inventory
- Some vendors first install and test at their facilities. This can complicate logistics but often improves the odds of the cluster working correctly.

Slide 92

Installation Planning – Site Details ✓

Information that you will need in order to plan installation with the vendor:

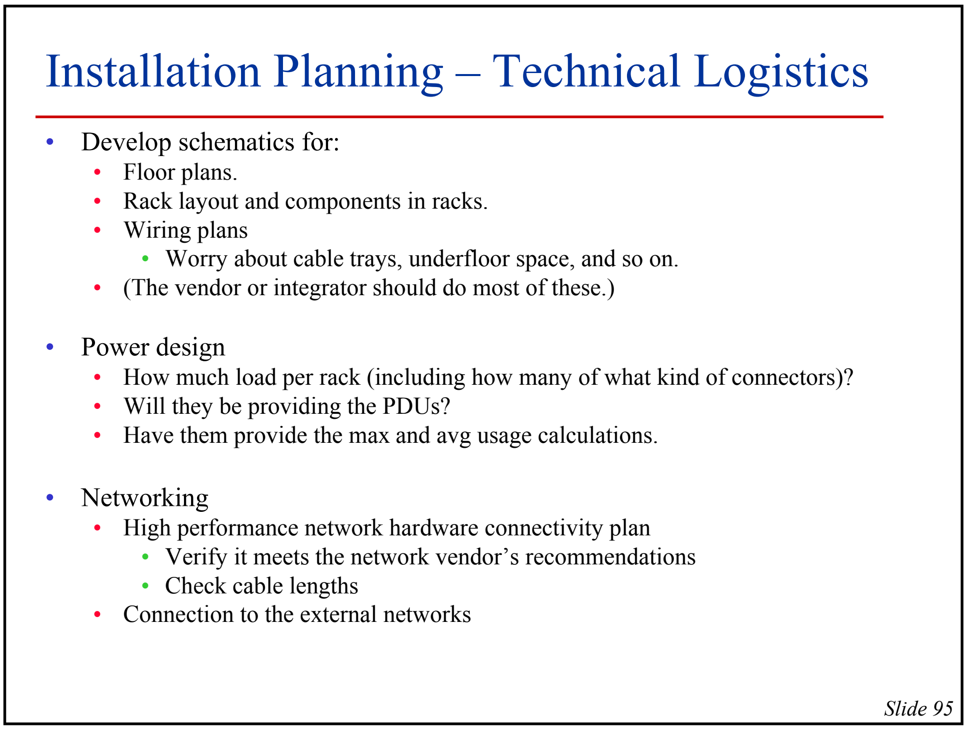
- Power
  - Estimate max (i.e. what happens when everything is turned on at once)
  - Circuits
  - What kind of connectors? If necessary, get pictures from the vendor.
  - PDUs
  - UPS
- Space
  - Floor layout
  - Floor tile cuts
  - Space under the floor
  - Staging and unpacking areas
- Cooling
  - Temperature
  - Air flow (capacity and potential under floor blockage)
- Delivery logistics
  - Can the truck get to the loading dock?
  - Will the crates fit in the doors?
  - Will you need to provide carts?
  - Will your boxes get lost if they're sent to Receiving?

Slide 93

Installation Planning – Vendor Logistics

- Have a designated vendor contact.
  - Status tracking, updates, notification, coordination of shipping.
- Develop a timeline.
  - Shipping, unpacking, installation, integration, testing.
  - How long will the vendor be involved?
  - How many people will they send?
  - Find out who will be in charge of the vendor team on site and check details with them before they arrive.
- Shipment and delivery plan
  - Are they shipping the cluster racked?
  - How many trucks?
  - What size trucks?
  - What will they need on your end?
  - What company?
  - If possible, a method to contact the drivers of the trucks once they are on the road should be arranged.

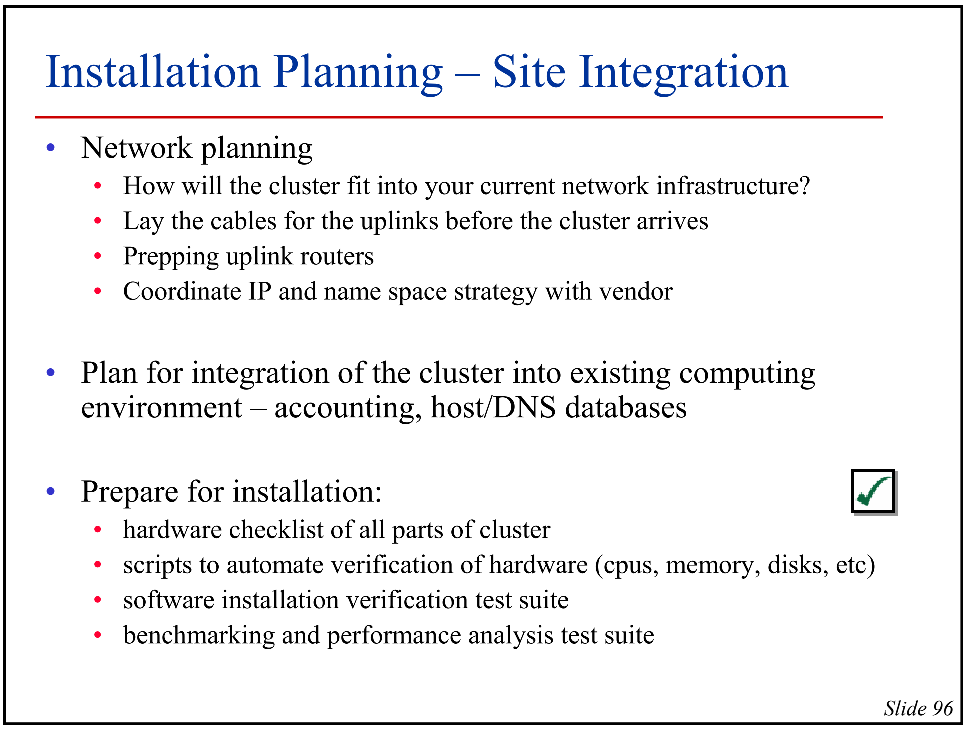
Slide 94



Installation Planning – Technical Logistics

- Develop schematics for:
  - Floor plans.
  - Rack layout and components in racks.
  - Wiring plans
    - Worry about cable trays, underfloor space, and so on.
  - (The vendor or integrator should do most of these.)
- Power design
  - How much load per rack (including how many of what kind of connectors)?
  - Will they be providing the PDUs?
  - Have them provide the max and avg usage calculations.
- Networking
  - High performance network hardware connectivity plan
    - Verify it meets the network vendor's recommendations
    - Check cable lengths
  - Connection to the external networks

Slide 95



Installation Planning – Site Integration

- Network planning
  - How will the cluster fit into your current network infrastructure?
  - Lay the cables for the uplinks before the cluster arrives
  - Prepping uplink routers
  - Coordinate IP and name space strategy with vendor
- Plan for integration of the cluster into existing computing environment – accounting, host/DNS databases
- Prepare for installation:
  - hardware checklist of all parts of cluster
  - scripts to automate verification of hardware (cpus, memory, disks, etc)
  - software installation verification test suite
  - benchmarking and performance analysis test suite

Slide 96

Installation Planning – Documents to Have ✓

- Facility
  - Floor Plan
  - Circuit maps
- Network diagrams
- Rack layout and wiring diagrams
- Component checklist with manufacturing model and part #s
- Task assignment list
- Timelines
  - Be realistic
  - Complete - from arrival, thru installation, thru acceptance testing, thru friendly user stage and into production
- Hardware and software product documentation

Slide 97

Arrival – The Big Day

- Arrival Day
  - Things are quite likely to arrive over a whole series of days, actually.

- The Critical Task – Inventory Checking
  - Boxes
    - Compare to shipping receipts
    - Inspect for damage
  - Contents
    - Check against the final parts list against final vendor quote.
  - Problems here can lead to serious problems between you, the vendor, and the shipping company. Good documentation will help immensely.
  - Keep a list of missing components, potential problems, and questions.
  - Digital cameras are useful if there is major damage to boxes.

Slide 98

Installation and System Check

- Your goal – get the system working well enough to check that each individual component has all the right parts.
  - Vendors or system integrators are likely to do installs.
  - They may not run tests on all the parts.
- Three major steps:
  - Physical installation
  - Initial configuration
  - System check
- The system check is important:
  - Catching problems that may slow progress towards production.
  - Reaching closure on the shipment.
  - Does not replace an acceptance test – just a first step.

Slide 99

Physical Installation

- Who does the installation?
  - You – which can be fun, or exhausting.
  - The vendor, with a team of installers.
- For large clusters, installation will take several days.
- Keep in mind that you will need to replace hardware, so:
  - Everything must be accessible and removable.
  - Everything, including wires, should be labeled.

Slide 100

Initial Configuration - Philosophy

- In order to run system tests, you must run commands on every system that check for correct and working hardware.
- Therefore, the cluster must be configured enough that you can run commands across the entire cluster.
- We will:
  - Start enough of the real configuration to enable tests.
  - Carry out the tests and address problems.
  - Continue the real configuration.
- Some people skip the system tests in this phase and carry them out later as part of the final cluster tests.

Slide 101

Initial Configuration – Steps

- Most of the details here are covered in the “Administration” section, thus we will just cover the basic topic list here.
- Carry out the initial configurations in each of these areas:
  - Simple network functioning.
  - Remote power and console working.
  - Develop an image to install on each node.
  - Set up the node booting mechanism.
  - Install the base image on all nodes.
- You can now:
  - Ping each node and get a response.
  - Ssh or rsh into each node as root.

Slide 102

System Check Mechanism

- Use a tool that allows you to run commands on all computers:
  - pdsh
  - C3 (such as in OSCAR)
  - many others
  - dumb serial shell script:

```
for h in $HOSTS; do
   ssh $h –c command
done
```

- On every system, check: ✓
  - CPU number
  - Memory installed and working
  - Disks installed and correct size
  - BIOS tests, if possible
  - Other hardware you expect (graphics cards, etc)
- Check network connectivity.
- Bring up issues with the vendor.

Slide 103

System Check – Software Inventory

Check the installed software:

- OS version
- Compilers, libraries, system software installed
- What else is there?
- Security patch levels

(Obviously, if you installed an image you developed yourself, then you will already know what's installed.)

Slide 104

Installation Lessons Learned, Part 1

- Expect the unexpected...
- Know the design plans inside and out.
- Be prepared to deal with problems
  - Can you get larger doors cut to fit the crates through
  - Can you rearrange the computer room to make more space?
  - Can you find replacement PDUs at a local store?
- If the vendor mentions that half your computers were assembled at a different factory from the other half, be worried.
- Actually look at every part and verify that it is what you are expecting.
- If you don't do a system check, you will discover that half your memory is occasionally bad a month after you've been in production mode.
- It ***always*** takes longer than you planned

Slide 105

Installation Lessons Learned, Part 2

- Know who has responsibility for which pieces, ensure that this is clear to you, your team **and** the onsite vendor techs.
- Have someone there with vendor at all times to
  - answer questions
  - make decisions and contingency plans
  - catch mistakes
  - document gotchas, tools for checking status of hardware and software, howtos, etc.
- Transition Event:
  - Plan for a day before the system is turned over, where you, your team and the vendor techs sit down and go over the system, hardware and software, management tools, etc. – get a brain dump from the techs before they leave
- Ask questions, don't assume that the techs know everything; if you see something odd, question it!

Slide 106

Schedule: Afternoon Part I

Slide 107

Afternoon, Part I

- System Configuration – Susan and Remy
  - A myriad of configuration and administration topics that must be addressed consistently when running a production cluster, from software installation to security.
- Resource Management – Bill
  - User job launching and execution, scheduling the system, and tracking usage.
- Applications and Environment – Pete and Bill
  - The software that the users see. Your options, and the issues associated with them.

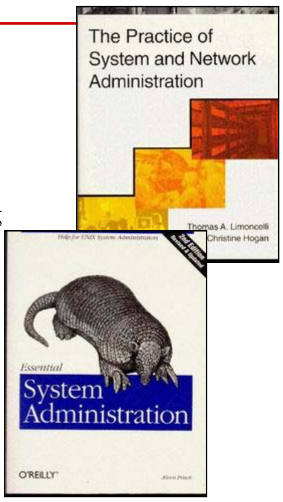
Slide 108

System Configuration

Slide 109

Two Primary Topics

- Cluster System Administration
  - Discussion of a number of important areas to understand in order to design, configure, and administer a production cluster.
  - We will **not** be discussing general systems administration, which is a pre-requisite for carrying out cluster administration.
    - A few favorite UNIX systems administration texts (see Appendix A):
      - Limoncelli & Hogan
      - Frisch
- The Configuration Process
  - Turning hardware in racks into a cluster that runs correctly.


The Practice of System and Network Administration
Thomas A. Limoncelli
Christine Hogan
Help for UNIX System Administrators
Essential
System Administration
O'REILLY

Slide 110

Philosophy: consistency and centralization

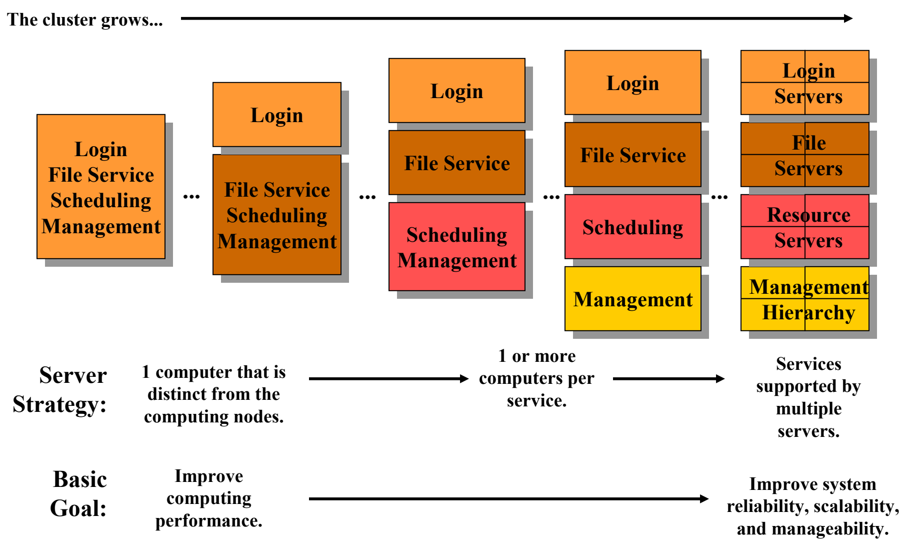
- Managing a lot of systems can be remarkably complex.
- Consistency, simplicity, and automation are critical.

- Ideally, when managing a cluster, you manage it:
  - As single system.
  - From a single point.
  - From an isolated management infrastructure.

- Let's look at a few configuration management approaches as an example of this philosophy.
  - We'll start with the concept of a dedicated management system.

Slide 111

Management Infrastructure



Slide 112

Management Approach: Ad Hoc

- Approach:
  - Load the OS onto each node by hand.
  - Make changes on individual nodes.
- Benefits
  - Easy.
  - Ok for small numbers of machines.
  - Natural for many sysadmins.
- Disadvantages:
  - Quickly leads to chaos and truly stunning debugging scenarios.
  - Doesn’t scale.

```
server #  rcp /etc/passwd node01
```

```
node01 #    wc -l /etc/passwd
10
```

```
node02 #    wc -l /etc/passwd
52
```

```
node03 #    wc –l /etc/passwd
/etc/passwd: No such file
```

Slide 113

Management Approach: Cloning

- Approach:
  - Have a standard OS build for your site.
  - Load that onto each node.
  - Ad hoc change management.
- Benefits
  - Fairly easy.
  - Better than pure ad hoc.
- Disadvantages:
  - Configuration drift - doesn’t handle changes over time, image gets out of date.
  - Doesn’t scale.

```
server #  rcp /etc/passwd node01
```

```
node01 #    wc -l /etc/passwd
10
```

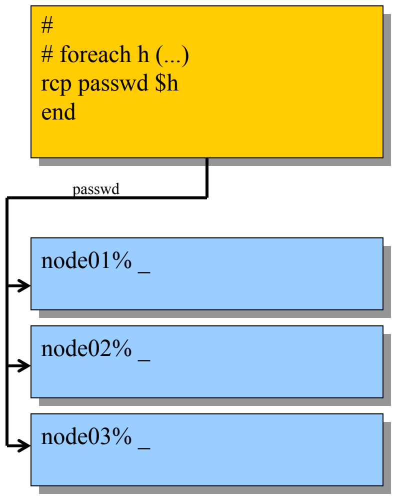
```
node02 #    wc -l /etc/passwd
10
```

```
node03 #    wc –l /etc/passwd
11
```

Slide 114

Management Approach: Centralize

- Approach:
  - Use cloning.
  - Have one system from where you can drive everything.
  - Make all changes from that system.
  - Generally do a loop across all nodes, making the same change everywhere.
- Benefits:
  - Not too tough.
  - Improves consistency.
- Disadvantages:
  - Doesn't handle down nodes or new nodes.
  - Some admins will tend to hack individual nodes anyway.

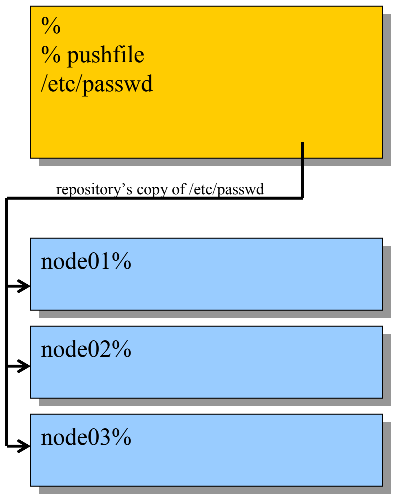


```
#
# foreach h (...)
rcp passwd $h
end
```

Slide 115

Management Approach: Image Database

- Approach:
  - Use centralized management.
  - Have a repository of node information which describes "how the world should be".
  - Use tools to get the nodes to match the database.
- Benefits:
  - Consistent environment.
- Disadvantages:
  - Requires fairly serious software infrastructure.
  - Requires rigor on part of admins.



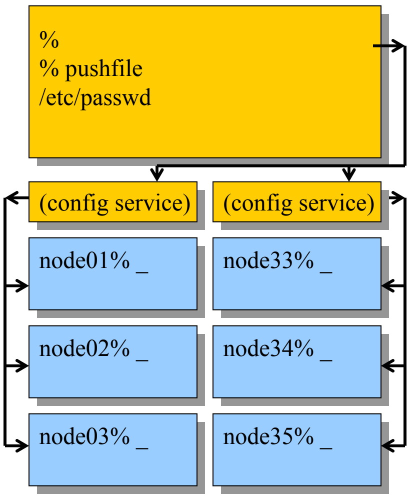
```
%
% pushfile
/etc/passwd
```

Slide 116

Management Approach: Hierarchical

- Approach:
  - Use repository management.
  - For every N nodes, have 1 system whose sole purpose in life is to make sure those nodes conform.
- Benefits:
  - Scales well to large systems.
  - Management systems can take on other responsibilities:
    - booting, file distribution, network services, ...
- Disadvantages:
  - Requires extra hardware and $.
  - Requires fairly serious software infrastructure.

- This is only necessary for very large or very complex systems.



Slide 117

Centralized Configuration Management

- So, with the centralized repository approach:
  - You can manage the entire system from one machine.
  - You make changes in one place, and they are propagated appropriately.
  - This model simplifies the administration.

- In reality, however:
  - You can manage *most* things from one place.
  - Individual servers often need hands-on work.
  - Occasionally a node will need direct attention.
  - Image management is quite complex.

Slide 118

OS Image Management

- What is Image Management?
  - Getting the OS to the node.
  - Configuring it for the first use ("build").
  - Configuring it over time.
- Which brings up some questions:
  - What is the OS?
  - Why does it need to be configured?
  - Why do we care?
  - What is the air speed velocity of an unladen swallow??

Slide 119

The Primary Issue is Change

- In an ideal world:
  - A new version of an OS...
    - ... would just work...
    - ... on all nodes, unchanged.
    - ... could, run, untouched, until the next OS release ...
    - ... or two.
  - You'd never need to...
    - ...add services.
    - ... install patches.
    - ... update network parameters.
    - ... debug problems.
    - ... install new apps.
    - ... wake up early.
- But in reality, depending on your environment and requirements:
  - The OS usually needs work.
  - You will have to make regular changes on your systems.
  - You'll need to support many different applications and libraries.
  - You'll have users who both want the same library, but want different versions of it.
  - Several people administer these systems at the same time.
  - You will have several different OS configurations installed as you will have different servers.
- OS Image Management is really about change management.

Slide 120

Node Things, Categorized by Change

Things on a Node		Example	Issue
OS	Base OS	Kernel, /, /root /usr, /bin, /lib, ...	Source: vendor Typically only changes with OS change.
	Base Mods	Replacements, links, permissions, ...	Source: admin One set of changes, made to base OS.
	Configs	/etc changes, crontabs, patches, ...	Source: admin Continual change for fixes, new features.
Applications		emacs, perl, mathematica, ...	Source: vendor & 3rd party Changes for new software, bugs, ...
User Space		Home directories, job input and output	Source: user Changes constantly

Slide 121

OS Image Management Strategy

1. Build the “base” system image.
   The OS release.
   Changes you have to make to it to get it to boot and be happy.

2. Have a reasonable way to make changes to the OS over time.
   The “configuration” system.

3. Eventually build a new base system image.
   Consider rolling in those changes.
      (It helps if you know what those were...)

Slide 122

Building the OS Image

- The "everything" approach:
  - Install base Linux distribution onto a system by hand.
  - Install other software, RPMs, and whatever else.
  - Make any changes to the OS.
  - Store the results as the "image".
    - Image type depends on how you plan to install it.
- Advantages:
  - Fairly easy.
- Disadvantages:
  - Not flexible.
  - Can be hard to debug weirdness.
- The "minimal" approach
  - Install base Linux distribution onto a system by hand.
  - Figure out how you want the final image to look.
  - Build a set of scripts and a list of RPMs that gets you to that image.
  - Store the result as an "image", and invoke the scripts as part of the build process.
- Advantages:
  - More likely to work on different types of nodes.
  - Easier to upgrade over time.
- Disadvantages:
  - Hard to get just right.

Slide 123

Getting the OS Image Onto Disk

- Diskless
  - Build it on the server.
- Incremental OS Load
  - Boot mechanism + Build Script + Network repository
    - Kickstart + RPMs
    - CD + RPMs or NFS
  - This is the standard way to build a standalone Linux workstation.
- Physical Disk Image
  - Basically a 'dd' of a working disk.
  - Handy for "other" operating systems.
  - Fairly easy to snag.
  - Very, very disk size specific.
- Logical Disk Image
  - Partition information for disk.
  - Files to put into those partitions.
    - tar, dd, or copy of net disk

Slide 124

Tools for Image and Configuration Management

- Cfengine - Mark Burgess
  - Maintain a class-based description of each system.
  - Each system runs cfengine and attempts to conform to description.
- System Installer
  - This is the standard tool for cloning, building all nodes from a "golden client". It doesn't handle ongoing change as well as cfengine. Many people use this in conjunction with cfengine.
- **Many** other tools in this space, some from vendors:
  - IBM: xCAT
  - Linux NetworX: Clusterworx

Slide 125

Additional Administrative Topics

- Now you have:
  - A management infrastructure.
  - A system for defining what bits should be on each computer and for getting them there.
- Some specific topics left to cover:
  - Booting
  - Naming
  - Accounts
  - Security
  - Monitoring
  - Backups

Slide 126

Booting

- In the near past, booting was surprisingly tricky:
  - Net booting didn't work reliably due to hardware variations.
- These days, the usual build/booting model is as follows:
  - Build an image on the master server, typically from CD.
  - The master server gets set up as a boot server using DHCP/TFTP.
  - Images are built on the master server for each node.
  - The other nodes in the cluster then boot using PXE from the boot server.
  - After that, nodes either boot from local disk or always from the boot server.
- Make sure your hardware can boot from the network.
- (If you can't, talk to us about hacking floppies or CDs to fake a network boot.)

Slide 127

Booting - Diskless or Local Disk

Diskless Booting vs. Local Disk

Diskless Booting

- The OS lives on a server and is NFS exported to the nodes.
- The nodes, if they have local disk, use it only for swap and temporary storage.
- Booting happens via DHCP/BOOTP and TFTP.
- Diskless booting is much easier to administer.
  - All management is done on the server, where the file systems of many nodes can be accessed simultaneously.

Local Disk

- A copy of the OS resides on the local disk.
- You have to get the OS onto that disk and maintain it.
- Local disk images are a bit more flexible.
- Local disk will generally perform better, depending on network, server, and app characteristics.
- Potentially more fault tolerant.

Slide 128

Address Assignment - Issues

- Option 1 - Hardcode the address in the OS image
  - Tricky.
  - The OS installation has to differ across nodes (undesirable), or ...
  - You must use tools that figure out what address to use.
    - (And these often need net access.)
  - May be necessary for some network types.
  - If you do hardcode it, use configuration management (later topic) to handle it.

- Option 2 - Assign the address via a network service.
  - Several options....

Slide 129

Address Assignment - Network Services

- DHCP - Dynamic
  - Pro: Easy
  - Con: You don't know which physical node has which hostname, and it will vary.

- DHCP - Static
  - Pro: You can find your hosts by name.
  - Con: Initialization is quite tricky because of the need to map ethernet addresses to IP addresses in the DHCP table. Ways to get the ethernet address include:
    - Type them all in
    - Staged/time-delayed boot
    - Read them on first boot, then build the database from that.
      - SNMP query network boxes
      - Serial line reading
  - Consider what happens when you replace some hardware.

- BOOTP
  - Basically a subset of DHCP... Still has that table initialization problem

Slide 130

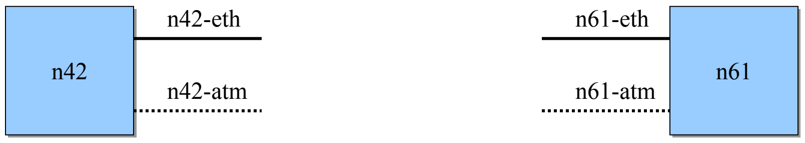
Subsequent Boot Mechanisms

- Your options are the same as the first boot, but:
  - You probably have DHCP and other servers configured.
  - You may have built your local disk during the first boot.

Slide 131

Naming - Issues

- Hostnames are important... Choose wisely.
- Hostnames actually refer to network interfaces.
  - Users need to use the right hostname to get the right network performance.
- One host will often have multiple hostnames.



- Some questions to ask:
  - What route does a packet from n42 to n61 take?
  - How can you change that?
  - What does “n42” mean in this case?

Slide 132

Naming - Recommendations

- Pick something you can type.
- Pick something you can auto generate.
- Be consistent.
- Don’t imbed too much information in names.
  - Names should be a hook to get that information from a database.
- Think about what the users need.
- Think about reverse name lookup.
- Think about what the routing issues will be.
- If possible, the generic node name should point to the fastest path in and out.
- Add cnames for personalized names.
- Document your naming convention.

Bad: c001t003r01n23
Bad: sparky, fuzzy01, fuzz02, server ...
Bad: node01, node2, ...
Bad: compute23_i386_2G_9G

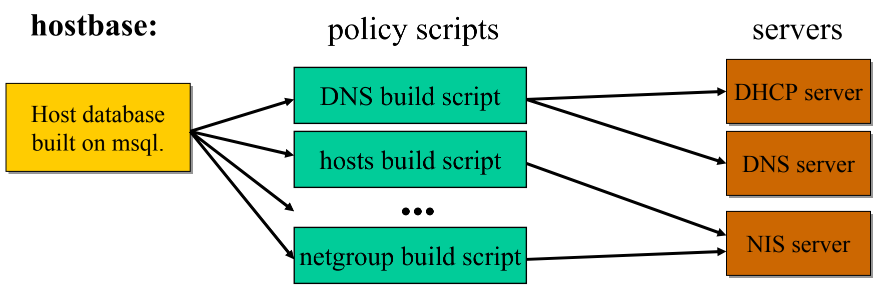
Probably: node1 - node32

I.e. node1 -> node-myr

“cluster” -> login1

Naming - Tools

- Hostname implementation is generally tackled at the site level, not the cluster level. (But this approach will work in either case).
  - Private networks complicate this.
- We highly recommend the use of a database backend to drive all hostname-related stuff.
- An example is “hostbase”, on the MCS systems tools web page



IP Address Ranges

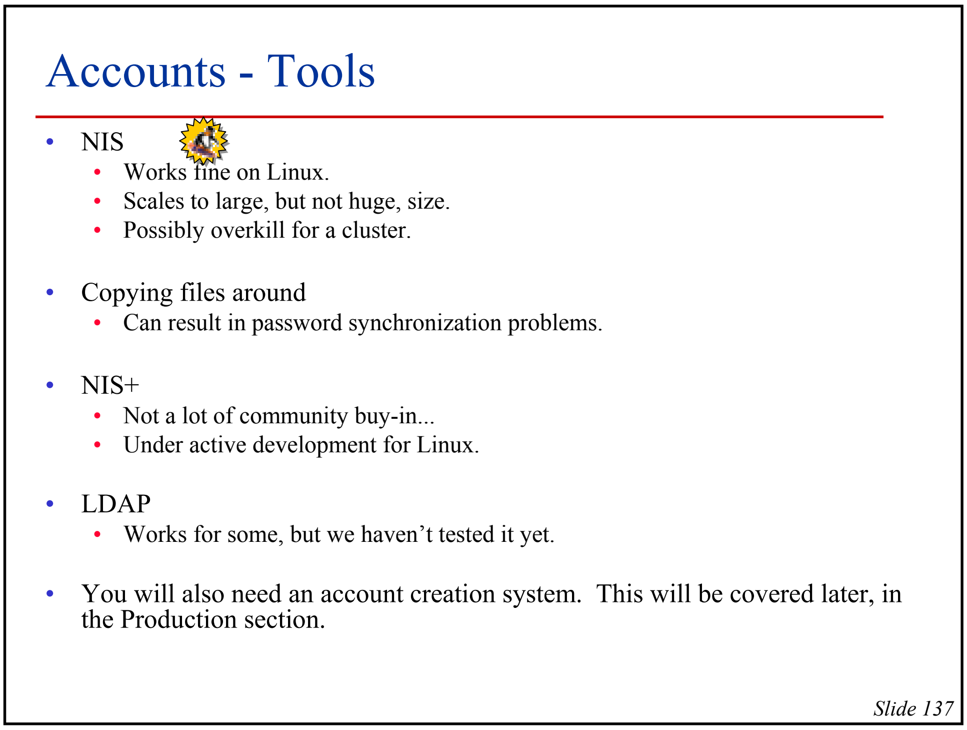
- For public networks, use whatever address ranges are available to you at your site.
- For private networks, use any of these address ranges:
  - 10.0.0.0 - 10.255.255.255
  - 172.16.0.0 - 172.31.255.255
    - We recommend this one... Lots of address space, and no need to do your own subnetting.
  - 192.168.0.0 - 192.168.255.255
- Before this, you'll need to have thought about routing and such, which will affect your address choices.
  - Avoid routing if you can, for speed and simplified management.

Slide 135

Accounts / Username Space

- You will need a way to distribute accounts across the cluster.
  - username <-> uid mappings
  - groups
  - home directories
  - passwords
  - ssh private keys
- Some of these are not so important on the computing nodes.
- The scheduler can/should handle some of this task.
  - In exclusive-mode use of the cluster, only one user has access to a node at a time. This is often implemented by modifying PAM access.conf file with prolog/epilog scheduler scripts.

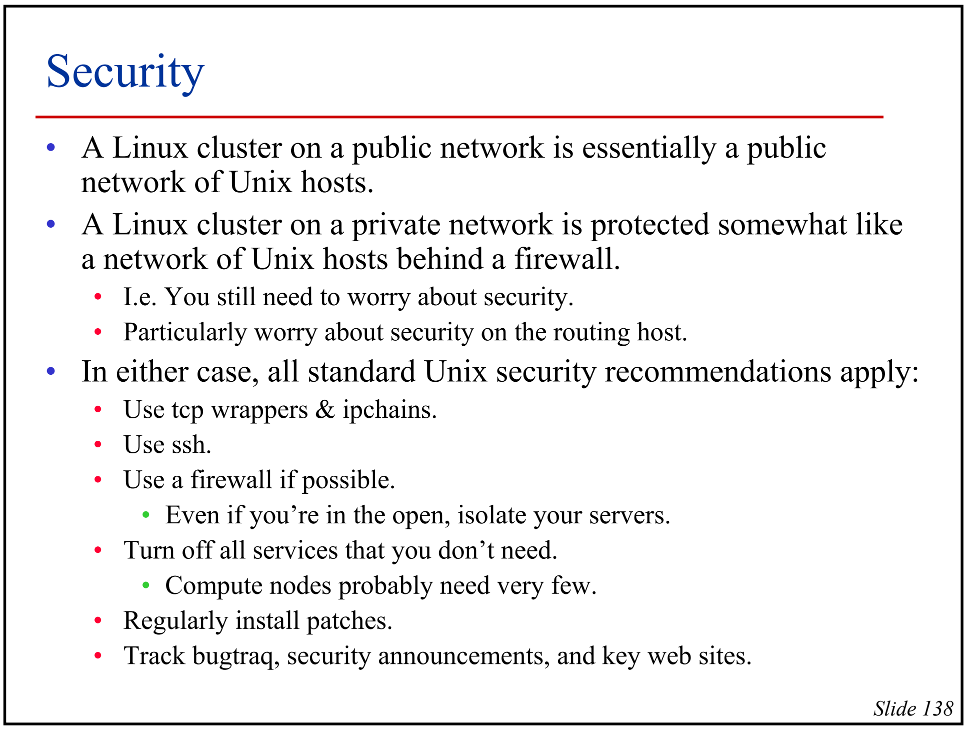
Slide 136



Accounts - Tools

- NIS
  - Works fine on Linux.
  - Scales to large, but not huge, size.
  - Possibly overkill for a cluster.
- Copying files around
  - Can result in password synchronization problems.
- NIS+
  - Not a lot of community buy-in...
  - Under active development for Linux.
- LDAP
  - Works for some, but we haven't tested it yet.
- You will also need an account creation system. This will be covered later, in the Production section.

Slide 137



Security

- A Linux cluster on a public network is essentially a public network of Unix hosts.
- A Linux cluster on a private network is protected somewhat like a network of Unix hosts behind a firewall.
  - I.e. You still need to worry about security.
  - Particularly worry about security on the routing host.
- In either case, all standard Unix security recommendations apply:
  - Use tcp wrappers & ipchains.
  - Use ssh.
  - Use a firewall if possible.
    - Even if you're in the open, isolate your servers.
  - Turn off all services that you don't need.
    - Compute nodes probably need very few.
  - Regularly install patches.
  - Track bugtraq, security announcements, and key web sites.

Slide 138

Monitoring, 1
Overview

- There are four parts to effectively monitoring your cluster:
  - Gathering (and managing) the data
  - Detecting the important bits
  - Notifying someone who cares
  - Taking corrective action
- Clusters, especially production clusters, provide some extra challenges and special opportunities.
- Most Linux distros provide simple logging and filtering (syslog plus logwatcher is a popular combo). These should not be used as is.

Slide 139

Monitoring, 2
Gathering and managing the data

- The biggest problem will be the volume of raw data, and we suggest you gather even more - standard system data (security alerts, system messages, /var/log/*) **plus**
  - Ambient temperature, airflow
  - On-board stats (cpus, fans)
  - Special hardware events (raid controllers, high-speed interconnects, etc)
  - Critical filesystem stats (home directories, /var on job mgt server, etc.)
  - Critical service stats (job mgt, web server, DNS, NIS, etc)
- Data gathering tools:
  - NAGIOS (Net Saint) - very, very flexible, steep learning curve
  - Ganglia – daemon running on each monitored node, easy to install
  - Clumon – monitoring system developed at NCSA
  - Supermon – interesting capabilities, minimal tools, kernel mods required
  - Home grown scripts – required with any of these tools

Slide 140

Monitoring, 3
Gathering and managing the data, 2

- Develop a strategy for dealing with the data, and do it consistently across the cluster and utilities
  - **local files** - you will need to deal with compressing and rotating the local files, be careful with cron
  - **central server** – standard syslog uses UDP, packets can get lost, switch to syslog-ng and set up tcp/ip connections (warning: really large environments will need to tier)
  - The different tools have different locations for storing their data, you will need to compress and rotate it, consider moving their data to the centralized server as well

Slide 141

Monitoring, 4
Notification & Corrective Actions

- How you notify will depend on the tool you are using.
- Most will have pager and email support.
- Use 3 levels of notifications:
  - Critical – users cannot work, fix immediately
  - Warning – users can continue to work, but it should be fixed soon
  - Informational – for tracking purposes
- Match with three tracked, email lists, if you use a pager, have the critical email list include the pager's email address.
- Monitoring is always a work-in-progress
  - For example, raid controller cards deaths due to fan failure causes 24 hr down time for storage server, add watcher for temperature events, critical notification level, call in upon first failure.

Slide 142

Monitoring, 5
Checklist

- Ongoing monitoring upkeep:
  - Gather data for specialized hardware.
  - Have consistent data management strategy.
  - Pick one of the core tools and incorporate data gathering, watchers, notifications and any automated corrective actions.
  - Set up watchers for all known failure modes.
  - Do continuous updating of failure patterns to the watchers.

Slide 143

Backups

- Systems backups:
  - You should back up the configuration system and data – whatever is necessary to restore the cluster in case of catastrophic failure.
  - What you should be backing up will depend on the configuration choices you made during the initial systems configuration.
  - The user documentation.
  - The user software environment.
  - Most of this data will be relatively static and relatively small.
  - Consider using mirrored system drives.

- User backups (Home directories):
  - Lots and lots of data, on the order of TB.
  - Can be highly dynamic.
  - Requires frequent backups.
  - This can seriously overload a regular backup system and can prove to be prohibitively expensive and prohibitively slow.
  - Consider using a hard drive backup system, with regular archival offsite.

Slide 144

The Configuration Process

- Previously Covered in “Purchasing and Installation”:
  - Get An Initial System Working
    - Enough to have nodes respond to ping.
  - Basic System Check and Inventory
    - Confirm that you have the right hardware and that it works.
- The next steps:
  - Fully configure the system.
  - Full-scale system testing.

Slide 145

Full System Configuration

- Tackle the major parts of the system in roughly this order:
  - Networks
    - Ethernet first, then the others
  - The primary server
  - Other servers
    - File servers, web servers
  - Standard system services
    - DNS, NTP, NIS, HTTP, etc
  - Configure file systems and file servers
  - Cluster-specific services
  - Tune the node images

Slide 146

System Testing

- After – or during – configuration, carry out extensive testing:
  - Outage Tests
    - Reboot everything. Does the cluster come back?
    - Yank the power on parts or all of the cluster. What happens?
  - Component Tests
    - Test each part of the cluster for functionality and performance.
  - Benchmarking
    - Run standard benchmarks.
    - Find out where you are on the Top500. (Your vendor will often do this for you if you have a large system.)
  - Stress Tests
  - Application Tests
    - Run your local applications to test performance and correctness.
  - Acceptance Tests
    - This is when you would run the formal acceptance tests, if any are part of your contract.

  - See Appendix B for more info on these.

Slide 147

Resource Management

Slide 148

Process and Resource Management

- How do we start jobs?
- Who decides where they are run?
- Who keeps users from getting in each other's way?

Slide 149

Running Jobs

For a production system, we want

- Space-sharing for MPI jobs
- Optional time-sharing for applications for which it makes sense (debugging MPI runs, parallel make, task farm)
- Utilization important: run jobs through the night; pack jobs.
- Complex scheduling policies
  - Fast turnaround for small jobs
  - Large jobs don't get starved
- Ability to run "interactive" (stdout connected to terminal) jobs in a space-shared environment
  - "mpirun -np 4 a.out" should work!
- All of this integrated with accounting

- The hostfile (specifying a list of nodes on which to launch a job) is suitable for single-user systems. Multi-user systems require a lot more (but vestiges of the hostfile remain).

Slide 150

Process Manager vs. Resource Manager

- A **process manager**
  - Starts the processes in an MPI application on some nodes
  - Directs output/error to the terminal
  - Propagates signals
  - May know what is running on the system (provides "ps" equivalent)
  - Examples: P4 (uses rsh), MPD, bproc, LAM
- A **resource manager**
  - Keeps track of requests and available resources
  - Decides what jobs run on what nodes at what time.
    - Makes policy decisions and asks process manager to implement
  - Examples: PBS, SGE, hostfile (scarily common)
  - A **scheduler** is part of a resource manager and sometimes is a modular component. It makes the decisions, but is told about available resources and constraints. Examples: Maui, Catalina

Slide 151

Process Managers

- P4 (built into MPICH/P4)
  - based on a machine file (list of node names); uses rsh/ssh
  - being replaced by MPD
- MPD
  - replacement for P4. Much faster, no rsh/ssh. Relies on MPD daemon running on each node. Distributed with MPICH but independent in theory.
- BPROC
  - Attempts to create a "single system image" – all processes show up in the process table of the master.
  - Integrated with MPICH in cluster software distributed by Scyld
- LAM/MPI
  - Full version of MPI integrated with process management

Slide 152

Portable Batch System

PBS

- Developed at NASA Ames Research Center for its supercomputers. Currently supported by Altair; Widely used.
- Two versions
  - Open PBS – open source
  - PBS Pro – costs $, improved features and stability
- Modular scheduler
  - Allows use of Maui or Catalina schedulers
- Fairly robust (but not perfect!)
- Can handle batch and interactive, space shared and time shared

Slide 153

Installing/Using PBS

- Installation/configuration is fairly complex.
  - Oscar and ROCKS automatically perform basic setup
- Submit/view jobs:
  - qsub –l nodes=4 –l walltime=5:00 <scriptname>
  - qstat –an
- Inside a job, ${PBS_NODEFILE} contains a list of nodes allocated to the job. Script runs on first node in the file
  - mpirun –np 4 –machinefile ${PBS_NODEFILE} a.out
  - Amazingly, no distributions provide a PBS-aware mpirun.
- PBS prologue/epilogue feature (not used by distributions) allows you to
  - clean nodes after a job
  - unlock/lock nodes so that users can’t log in when they don’t “own” a node

Slide 154

External Schedulers: Maui and Catalina

- The default OpenPBS scheduler (FIFO) is not sophisticated and will not effectively utilize resources
- The scheduler in PBS PRO is much better
- Maui
  - Full bells-and-whistles 3rd party scheduler (can also be used as a full resource manager, but usually used with PBS)
  - Stability problems make it painful to use
  - In use in many places
  - Download from Sourceforge.net
- Catalina
  - NPACI replacement for Maui. Written in Python instead of Java
  - Less functionality. More Grid oriented
  - Not offered for stable production use (in active development)

Slide 155

PBS Recommendations

- OpenPBS (+ Maui) works well for small (<= 32 node) clusters with moderate usage.
- OpenPBS has stability and scalability issues with larger, high-use clusters. In our experience, Maui is more likely to break than OpenPBS.
- PBS Pro addresses these issues and also has a much more sophisticated scheduler. We recommend that you strongly consider paying $ to get PBS Pro if you have a large production cluster.
- There is an effort to fix OpenPBS – Scalable PBS at supercluster.org. Under active development.

Slide 156

Sun Grid Engine

- SGE is available from Sun under Open Source License
  - Standard Edition is for scheduling jobs on clusters
  - “Enterprise Edition” (not open source) is for clusters of clusters (Grid computing)

- Robust feature set. Similar command set to PBS
- Appears to be oriented toward load balancing of serial jobs. Parallel job support looks thin.
- Does not generate accounting logs. “qacct” output is oriented towards serial jobs.
- Not yet widely adopted, so experience is limited.
- ROCKS will make SGE the default scheduler in the next release.

Slide 157

Other process/resource managers

- Quadrics
  - If you have a quadrics interconnect, you have to use Quadrics Resource Management System (RMS)
- LSF
  - LSF (Load Sharing Facility) is commercial software. Very expensive, and has not taken hold in the parallel computing community.
- Myrinet
  - Myrinet applications need special initialization, which is handled automatically by Myrinet “process manager” code in both MPICH and LAM versions of MPI.

Slide 158

Accounting

- Clusters cost lots of $. People who spend the $ may want to know that the cluster is being used appropriately.
- "Accounting" = usage tracking + allocation management
  - Usage tracking: how much time was used, by whom, when
  - Allocation management: who gets to use the machine, how much time to they have, etc. Don't allow someone to run if they're out of time.
- Usage tracking is the easy one.
  - PBS provides detailed accounting logs, but does not have tools to summarize this accounting data. You will need to write your own.
    - Perl is your friend
    - You will probably want to load raw data into a real database and analyze it there.
  - SGE does not even generate good accounting logs.

Slide 159

Allocation Management

- Allocation management is difficult.
  - What you really want is seamless integration of allocation management with the resource manager (particularly with the scheduler).
  - Unfortunately, this is impossible to find. Some resource managers, such as LSF, provide some functionality, others, such as PBS Pro, provide none.
- The best available tool is QBank
  - Developed at PNNL
  - In use on Jazz cluster at ANL
  - Works with PBSPro, OpenPBS, LoadLeveler
  - Unless you use Maui, which has special hooks to integrate with QBank, you will be required to write many scripts, and the allocation system will be very fragile.

Slide 160

Applications and Environment

Slide 161

Applications and Environment

- Programming models
  - Most common parallel prog model: SPMD with message passing or put/get library.
  - Others: parameter studies (hundreds or thousands of small, independent runs)
- Development Environment
  - Compilers
  - Libraries
  - Debuggers
- Tools
  - Supporting multiple versions of software
  - Performance analysis

Slide 162

IPC For Parallel Programs

- MPI
  - Industry standard. Your code will work everywhere.
  - Allows scalable and high performance programs
  - Designed to be integrated into production environments
- PVM
  - Not a standard
  - Performance limitations
  - Designed for “personal parallel computers” – difficult to integrate into a production system as it includes an “operating system”. We recommend against except for special applications/environments.
- Threads/OpenMP
  - Can be used within an SMP node, but does not remove the need for MPI between nodes. Mixed-mode programming is fine if it works for you.
- Virtual Shared Memory
  - No standards. Largely negative experiences reported. Rarely seen today.

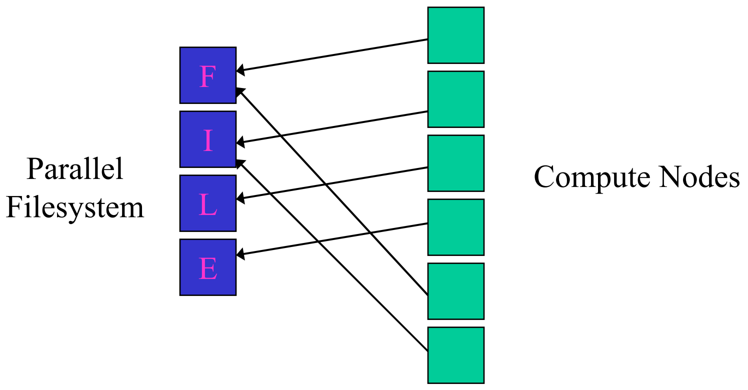
Slide 163

MPI Implementations

- MPICH from ANL
  - Different transport layers available (TCP, Myrinet)
  - In wide use because of its “abstract device interface” that makes it relatively easy to port to new network architectures and its historical importance to MPI
  - Includes mpd, a process management daemon
- LAM/MPI from Indiana University
  - Large feature set (e.g. most of MPI-2), runs on most hardware, excellent performance
  - Good documentation/info/web site
  - Excellent support
  - Not as widely used as MPICH for historical reasons, but check it out!
- MPI/Pro from MPI Software Technologies
  - Commercial product. Good support, performance, but not clearly better than LAM.
  - Costs $. Consider if you need to have commercial support. Also best solution for Windows-based clusters.

Slide 164

Parallel I/O



Slide 165

MPI-IO and ROMIO

- Doing efficient parallel I/O is tricky.
- Need an efficient software interface *and* a parallel/scalable filesystem.
  - Problems with Posix file symantics – why NFS can't be an efficient parallel file system
- MPI-2 I/O is the standard interface for cooperative parallel I/O
- The best/only implementation for clusters is ROMIO.
  - Works with both MPICH and LAM
  - Best of class:
    - IBM GPFS for Linux (only available to those who buy IBM hardware)
    - PVFS: Free and fast
    - NFS: ROMIO does have a way to use NFS, but the performance will be very poor.

Slide 166

Compilers

Expect to buy one if you want really good performance

- Open source
  - gcc/g++ performance is decent
  - g77 works, but cannot compete well with commercial varieties
- Portland Group
  - C/C++/Fortran 90 compilers
  - OpenMP (for automatic parallelization within a node)
  - Intel/AMD supported
- Intel
  - C/C++/Fortran 90 compilers
  - Intel only (no AMD) of course. Trial version available.
- 64-bit
  - AMD put a lot of effort into getting gcc to work well for the AMD x86-64 platform

Slide 167

Scientific Libraries/Tools

- For a excellent comprehensive list, see the NCSA HPC software page
- Some well-known and useful examples:
  - BLAS by Kazushige Goto
    - Available for P3/P4/Itanium/Opteron
  - BLAST
    - Computational biology programs
  - FFTW
    - Fast Fourier Transforms (Parallel)
  - PETSc
    - Parallel solution of PDEs
  - Commercial scientific libraries
    - IMSL
    - NAG

Slide 168

Debuggers

Slim pickings here.

- Totalview is the best parallel debugger
  - Expensive
- The best serial debugger is the Data Display Debugger (DDD). Lam has support for using DDD with LAM applications.

Slide 169

Performance Profiling

Tools for investigating the performance of parallel programs

- Jumpshot
  - communication profiling for MPICH applications
- XMPI
  - communication profiling for LAM/MPI applications
- Performance API (PAPI)
  - API for accessing hardware counters
- PerfSuite
  - Tools for performance analysis on Linux platforms
- Tau - Tuning and Analysis Utilities
  - Sophisticated code instrumentation

Slide 170

Installing Multiple Versions of Software

- Problem: experience has shown that software upgrades often have regressions.
- A production computing environment should allow the user to choose amoung different versions of software to make upgrades run smoothly
  - Compilers
  - MPI libraries

- RPM does not support multiple installed versions.
- Tools you can use to support multiple versions
  - softenv
  - modules
  - switcher: included as part of OSCAR

Slide 171

Schedule: Afternoon, Part II

Slide 172

Afternoon, Part II

- HPC Distributions – Susan and Remy
  - Your options for the software suite to put on the cluster, ranging from do it yourself to vendor-supplied.
- Serious Production Mode – Susan
  - What it takes to run the system in support of a complex user community.
- Collecting It All Together – Pete
  - All of these topics have interdependencies. Here's how everything fits together.

Slide 173

HPC Distributions

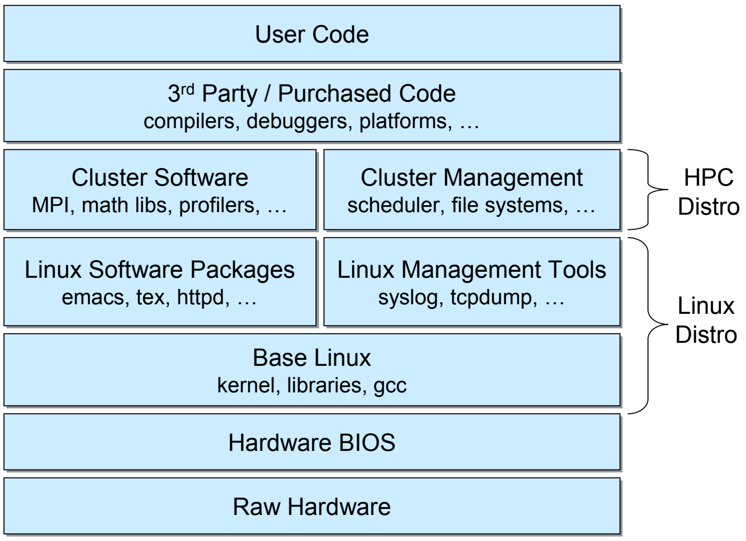
Slide 174

The Cluster Software Environment

- What software will you install on your cluster, and how will you install it?
  - Finally… the last of the five major decisions.
  - Many implications.
  - Many options.
  - Should be driven by these requirements:
    - What will your cluster be used for?
    - How much expertise do you have available to manage the cluster?

Slide 175

The Cluster Software Stack



Slide 176

Cluster Software Options

- Do It Yourself
  - Install your favorite Linux, then add the HPC packages you need.
- HPC Distribution
  - Use one of several pre-packaged distributions that includes many popular HPC tools on top of a Linux distribution.
- Vendor Distribution
  - Use whatever the vendor provides, which may range from nothing to an augmented HPC distribution.

Unfortunately, no solution is complete. We will "discuss do-it-yourself" in order to understand the full scope of the issue, then examine popular distributions.

Slide 177

Linux Distributions

- Top 3: RedHat, SUSE, Debian
- All three are stable, robust, complete.
- Consider what you are running on all your other machines, whether you want support from a commercial company, your political leanings.
- You will also need to pick the version - check what any 3rd party applications might require.
- If you use one of the HPC packages, your choice will be dictated.
- The two critical choices are distribution-independent
  - which packages to install (for example, development tools)
  - which kernel to use

Slide 178

Common HPC Kernel Modifications

- The current stable Linux kernel (v2.4) is fairly good for HPC.
- Common modifications/modules
  - High performance networking support – all high performance networks have drivers that are not included in standard Linux distributions
  - Modifications for specific cluster software
    - Totalview
    - Lustre and other filesystems
  - Crash dump facility (useful for cutting-edge systems)
  - Modifications to increase resources
    - Increase max number of file descriptors from 1024 to 8192
  - Modifications to provide useful additional functionality
    - Hardware performance counters
    - Stafs64 system call
    - Hardware monitoring (e.g. P4 thermal sensor trip)
    - Report ECC correctable errors and panic on uncorrectable errors
  - Other hardware support
    - Even for a common hardware, there are often new patches available that dramatically improve stability or performance.
- See the LLNL CHAOS project for more information on many of these.

Slide 179

HPC Distributions

- We evaluated and tested two open source HPC distros:
  - OSCAR
  - Rocks
- We analyzed one HPC distro product:
  - Scyld Beowulf
- Other distributions exist:
  - Mandrake's CLIC
  - LANL's Clustermatic
  - Sandia's Cplant
  - .. and others

Slide 180

Related Software, but not HPC Distributions

A number of software packages are often mentioned in the context of clusters but are not HPC distributions supporting parallel computing. We mention a few of these here in case they may be useful to you, but we will not discuss these in any detail.

Cycle Harvesting – using unused desktop cycles to carry out computation. Can usually also run on dedicated clusters for batches of individual jobs.

- Condor
- Entropia

Process Migration Systems – often include load balancing facilities

- MOSIX
- OpenMOSIX

Grid Computing – tools for coordinating distributed resources, often clusters. Usually layered on top of existing cluster software.

Our HPC Distro Evaluation Methodology

- Acquire latest release of distro.
- Install:
  - Install on 4-node cluster (1 master, 4 compute), IA32, Ethernet, Myrinet.
- Test Application Environment:
  - Run HPL and other simple application tests.
  - Check out user environment and tools.
- Test Administration Environment:
  - Add a node to the cluster.
  - Add software to the cluster.
  - Change the configuration of PBS.
- Evaluate Cross-cutting issues:
  - Documentation
  - User Support
  - Consistency
- Bottom line:
  - Would a newbie be able to get to a working cluster without becoming an expert?
  - Would an experienced admin find this useful?

OSCAR Datasheet

- Main site: http://oscar.sourceforge.net/
- Authors / Participants:
  - Oak Ridge National Laboratory
  - National Center for Supercomputing Applications
  - Dell
  - IBM
  - Indiana University
  - Intel
  - Bald Guy Software
  - Sherbrooke University
  - Ericsson
  - MSC.Software
  - Many other contributors
- Latest release: 2.3.1, September 2003
- Works with: Red Hat 8.0, Red Hat 9.0, Mandrake 9.0
- Hardware supported: IA32, IA64 (in older OSCAR versions), Ethernet, Myrinet
- Cost: Free
- License: GPL
  - Distro includes many flavors of open source packages
- Support model: Mailing lists
  - Resources: installation guide, users guide, mailing list archives, active mailing lists

Slide 183

OSCAR Philosophy

- Stated goal: "OSCAR version 2.3.1 is a snapshot of the best known methods for building, programming, and using clusters."
- Key philosophical issues:
  - OSCAR is largely an integration effort:
    - OSCAR collects a lot of software packages and integrates them.
    - The focus is on installation. The user has many installation options that are covered well in the documentation.
  - Architecture:
    - Standard cluster model with batch and interactive job support.
    - One master/login/server node, N compute nodes.
  - Management:
    - All management is done on the master.
    - No configuration model is enforced. SIS can be used to resync nodes.
  - Organization:
    - A SQL database is used to organize cluster information.

Slide 184

OSCAR Components

- Administration/Configuration
  - SIS, C3, OPIUM, Kernel-Picker, NTPconfig cluster services (dhcp, nfs, ...)
  - Security: Pfilter, OpenSSH
- HPC Services/Tools
  - Parallel Libs: MPICH (P4/GM), LAM/MPI, PVM
  - OpenPBS/MAUI
  - HDF5
  - Ganglia, Clumon
  - Other 3rd party OSCAR Packages
- Core Infrastructure/Management
  - System Installation Suite (SIS), Cluster Command & Control (C3), Env-Switcher,
  - OSCAR DAtabase (ODA), OSCAR Package Downloader (OPD)

Slide 185

OSCAR Experiences

- Installation
  - We had many problems, including mysterious crashes, building problems, and configuration errors.
  - The documentation was incorrect in some details.
  - Solutions to some of these problems were in the mail archives.
  - Solutions to others required extensive debugging, guesswork, and by-hand modification of configuration files.
- Application
  - Application tests ran fine.
- Administration
  - Adding software and nodes worked as described in the documentation.
  - PBS configuration was possible but required expertise beyond doc scope.
- Cross-cutting Issues
  - The documentation is extensive, but if you have problems, the docs are not as helpful as necessary.
  - The mailing list support is good.

Slide 186

OSCAR – Bottom Line

- Would a newbie be able to get to a working cluster without becoming an expert?
  - Yes, if their hardware and installation choices triggered no installation problems.
  - Yes, if they went to the mailing list with problems.
  - From mailing list traffic, it’s clear that new cluster admins do use OSCAR.
- Would an experienced admin find this useful?
  - Yes, as a way to jump past doing configs of various packages.
  - However, on a complex cluster:
    - The admin will then need to become an OSCAR expert.
    - The admin will eventually need to modify these packages.

Rocks Datasheet

- Main site: http://www.rocksclusters.org/
- Authors:
  - Rocks Cluster Group, San Diego Supercomputer Center
  - Contributors:
    - Millennium Group, UC Berkeley
    - Linux Competency Centre, SCS, Singapore
    - OpenSCE, Thailand
    - Many additional supporters and contributors
- Latest release: 3.0.0 (Lhotse), September 2003
- Based on: Red Hat 7.3
- Hardware supported: IA32, IA64, Ethernet, Myrinet.
- Cost: Free
- License: Open Source
  - Freely redistributable, copyright notice must be retained, etc
  - Distro includes many flavors of open source packages
- Support model: Not explicitly stated
  - Resources: users guide, mailing list archives, active mailing lists

Rocks Philosophy

- Stated goal: “Make clusters easy” – to deploy, manage, upgrade, and scale. There is a focus on building a “cluster appliance”.

- Key philosophical issues:
  - Architecture:
    - Standard cluster model with batch and interactive job support.
    - One master/login/server node, N compute nodes.
    - A specific cluster architecture is strongly recommended, i.e. an internal private network is expected, the master should have 2 NICs, and so on.
  - Management:
    - All management is done on the master.
    - Changes to compute nodes are handled by re-installation to maintain consistency. Other models such as cfengine can be overlaid.
  - Organization:
    - A SQL database is used to organize cluster information.
    - Rocks uses a configuration graph on top of RedHat Kickstart to optimize image management.

Slide 189

Rocks Components

- On by default:
  - RedHat 7.3
  - OpenPBS
  - Maui
  - MPICH (P4/GM/MPD)
  - Ganglia
  - NIS for authentication
  - NFS for sharing home directories
  - Web interface to cluster information
  - Rocks package and configuration management tools

- Included Options:
  - Sun Grid Engine
  - 411 as a secure replacement for NIS
  - Grid stack (globus, pgt, condor-g, kx509)

Slide 190

Rocks Experiences

- Installation
  - Our cluster network configuration did not match the Rocks design. This caused problems that we had to resolve manually.
  - The installation document was generally good, with many examples.
- Application
  - The user environment is nicely designed. For example, SSH is auto-configured for the user.
    - However, we ran into some problems. The SSH configuration didn't work correctly. (And is it necessary on a small internal network?)
  - Application tests ran fine.
- Administration
  - Adding software and nodes worked as described in the documentation.
  - PBS configuration was possible.
  - It's clear that the SQL database is essential (and sensitive).
- Cross-cutting Issues
  - The documentation that exists is very good, but many topics are left untouched, e.g. the SSH configuration, PBS configuration, and using the SQL database.
  - The mailing list support is good, but using it is essential.

Slide 191

Rocks – Bottom Line

- Would a newbie be able to get to a working cluster without becoming an expert?
  - Yes, if their system meets the requirements, i.e. has the required network architecture.
  - From mailing list traffic, it's clear that new cluster admins do use Rocks.

- Would an experienced admin find this useful?
  - Yes, if the management model fits their environment.
  - In particular, the Rocks Kickstart Graph and the SQL database could be powerful tools for a sophisticated cluster, but this would require that the admin become a Rocks expert.

Slide 192

Scyld Beowulf Datasheet

Scyld Computing Corporation™

- Main site: http://www.scyld.com/
- Authors:
  - Scyld Computing Corporation
- Latest release:
  - Scyld Beowulf Professional Edition (28cz-6), October 2003 (2.4.22 kernel)
  - Scyld Basic Edition – quite old
- Based on: Red Hat 7.2/7.3
- Hardware supported: IA32, Ethernet, Myrinet
- Cost: $375/processor (min 4 processors)
- License: Closed
  - Scyld Beowulf itself is “no redistribution, one archival copy”, and so on
  - The distro includes many flavors of open source packages which are still open
- Support model:
  - Can purchase training courses.
  - Can purchase documentation, and it is included in the professional edition
  - Basic documentation exists on the web.
  - Mailing list support.

- All information on this and related slides refers to the Professional Edition. E.g. the basic edition has only copyright and trademark restrictions, but has not been updated recently.

Slide 193

Scyld Beowulf Philosophy

- Stated goal: “a re-architected, second-generation Beowulf system designed for easy deployment, simplified long-term administration and higher performance.”

- Key philosophical issues:
  - Architecture:
    - The compute nodes run a minimal OS, with kernel modifications supporting process migration.
    - Users run jobs on the front end, which are then migrated to the compute nodes.
    - The result is a single process space across a distributed-memory cluster.
  - Management:
    - All management is done on the master.
    - The compute nodes require very little, if any, management.
    - Addition of new compute nodes is trivial.

Slide 194

Scyld Beowulf Components

- Included in a Scyld Beowulf distribution:
  - Scyld install and management tools
  - MPICH 1.25 (P4/GM)
  - PVM
  - PVFS
  - Scalapack
  - HPL
  - Lmsensors
  - BProc configuration management, monitoring tools
- There may be other components we missed – see next slide.
- Commercial products – such as PBS Pro, Totalview – are available for Scyld Beowulf.

Slide 195

Scyld Beowulf Experiences

- Overall
  - We tested the Professional Edition, but not extensively, thus our results are inconclusive.
- Testing
  - We were able to install the software (with minor hitches) and run test applications.
  - We did not extensively test the management or process model.
- Assessment
  - This is an interesting approach to parallel computing and HPC distribution.
  - If the computing model works for your applications and a commercial solution is acceptable, we recommend that you consider this, largely because the model does simplify administration.

Slide 196

Recommendations

- All of these packages enable cluster computing and are working for real users. Your best choice depends on your situation.
- OSCAR collects a number of packages together, allowing the administrator to make many decisions. It is appropriate for:
  - … new administrators with complex requirements.
  - … experienced administrators who want to bypass basic configuration.
- Rocks enforces a particular system architecture and administration model in order to simplify cluster management. It is appropriate for:
  - … new administrators.
  - … administrators who want things to work without getting into details.
  - … administrators who appreciate the coherent model and want to build on it.
- Scyld dictates a single-system image, simplifying administration as a consequence. It is appropriate if:
  - … all of your apps and users can use the single system image.
  - … you have the budget for the software.

Slide 197

Observations

- None of these distributions provides a solution for large-scale production by itself. They lack:
  - Large-scale I/O architecture
  - Allocation and project management
  - Differentiated system types
- All of these could be used as a basis for such a system.
- If using any of these systems on a complex production cluster, the administrator will eventually need to become an expert in that distribution itself, in addition to the individual packages.
- The expertise available on the mailing lists can be an excellent resource (for more than just distro-specific questions).

Slide 198

Serious Production Mode

Slide 199

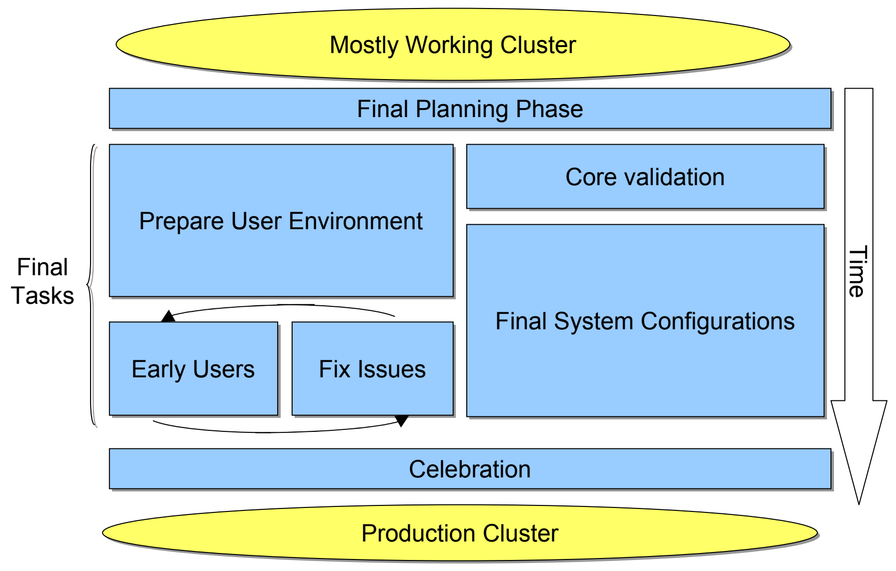
The Home Stretch

- The cluster’s up and mostly functional.
- The software’s installed.
- Acceptance tests are completed.
- There are no big technical unknowns remaining.

- So, what’s left?
  - Preparation for production mode.
  - Transition to production mode.
  - Production operations.

Slide 200

Getting to Serious Production Mode



What makes a cluster ‘Production’?

A production cluster is one with these guiding principles:

✓ **Users come first**

Every decision answers these questions first:

‘how does this help the users’

‘how will this affect the users’

✓ **Always optimize for uptime**

Stability always wins over performance or new features

✓ **Tightly control the environment**

Strong change management policies

No random changes, no long-term hacks

Simple & Coherent

In essence, the term “production” sets expectations.

Slide 202

Planning – Developing a timeline

- Time estimates:
  - Give yourself at least 2 months for the early user phase.
  - Keep an eye out for potential gotchas such as stalled tasks which often have hidden problems.
  - Estimate high for building infrastructure for user resources and support.
  - Consider starting simple to save time - a simple scheduling algorithm, do usage data tracking w/o a full allocation mgt package, etc.
- Choose and announce a formal production date.
  - Be aware if your environment allows schedule slippage or not…
- Schedule staff status update meetings at all important decision points - consider weekly meetings thru the first months of production.

Slide 203

Planning - Prioritizing

- Prepare a list of "Go/No Go Production" criteria –
  - i.e. if the scheduler continuously dies, there is no point in attempting to go production.
  - If any one of these aren't met, the transition to production cannot happen.
  - This list defines your top priority items.
- Order the remaining tasks by considering how the task relates to the three guiding principles.
  - Users, stability, consistency.
- But…. be careful, don't short time on tasks that will make management of the cluster easier.

Slide 204

Doing the work

- After the planning phase, there is a lot of work that can be done in parallel.
- For simplification, we will group the remaining tasks into two categories:
  - System preparation
  - User preparation
- We will talk about the remaining system tasks first, then cover preparing for the users.

- Once these have been completed, you will be ready to transition to official production.

Slide 205

Final System Preparation – Step 1
Sanity Check

- Before starting the final system configurations, double check that the system is at reasonably function level.

- Using the mechanisms discussed in the **System Configuration** section, verify that you can perform core administration tasks:
  - **Add a user.** Did their password entry and home directory get set up properly?
  - **Fail and replace a node**. Did the replacement gets the right image and right network configuration?
  - **Remotely power cycle a set of nodes**. Did they all come up cleanly? Could you watch them boot on their consoles?

- Confirm that usage and security policies are implemented consistently across the cluster.

Slide 206

Final System Preparation – Step 2
Remaining System Configurations

- Finish the monitoring systems (see "configuration" section)
- Confirm backups are working (see "configuration" section)
- Failover services for DNS, NIS, etc
- Final Check

Slide 207

Failover services for DNS, NIS, etc

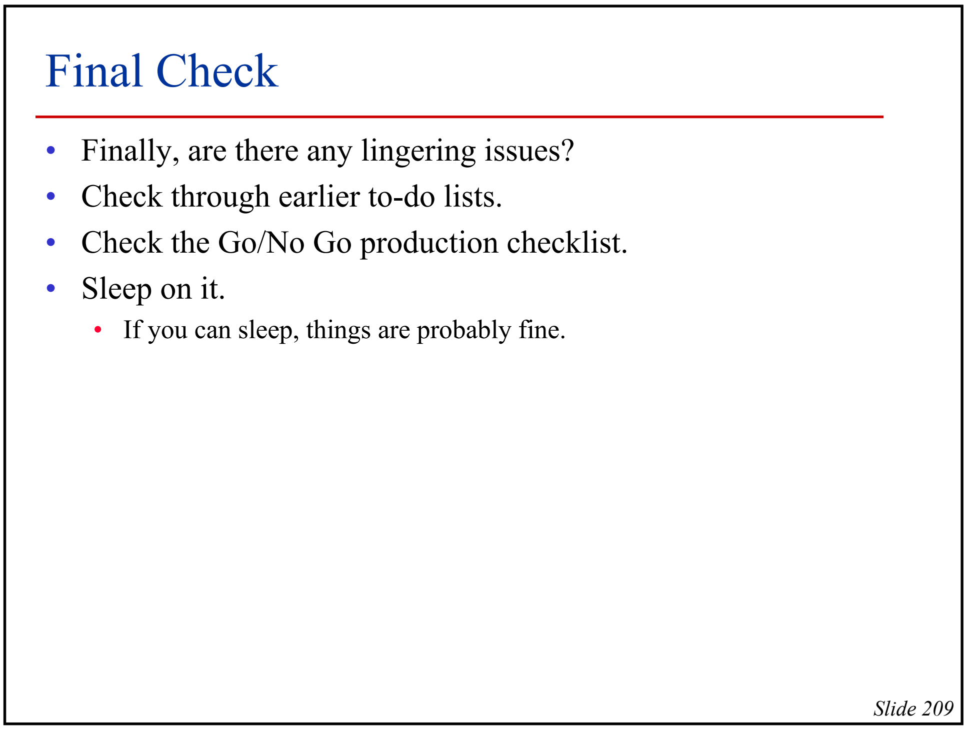
- Protection for critical services
  - If possible, provide all services on the local network..
    - The goal is to have the cluster to continue to work even if all external network access is lost.
  - Choose one or more for each critical service:
    - Redundant servers, if one fails the service doesn't fail.

      7 of the 8 Jazz global file servers can die and home directories will still be accessible, even if the access is very slow.
    - Replacement servers, unused but instantly ready

      At least one front-end server is not in production use, ready to be added to the rotation if an active one fails.
    - Spare parts, especially for parts with frequent failure rates.

      We keep extra Jazz PVFS server power supplies due to 30% failure rate.

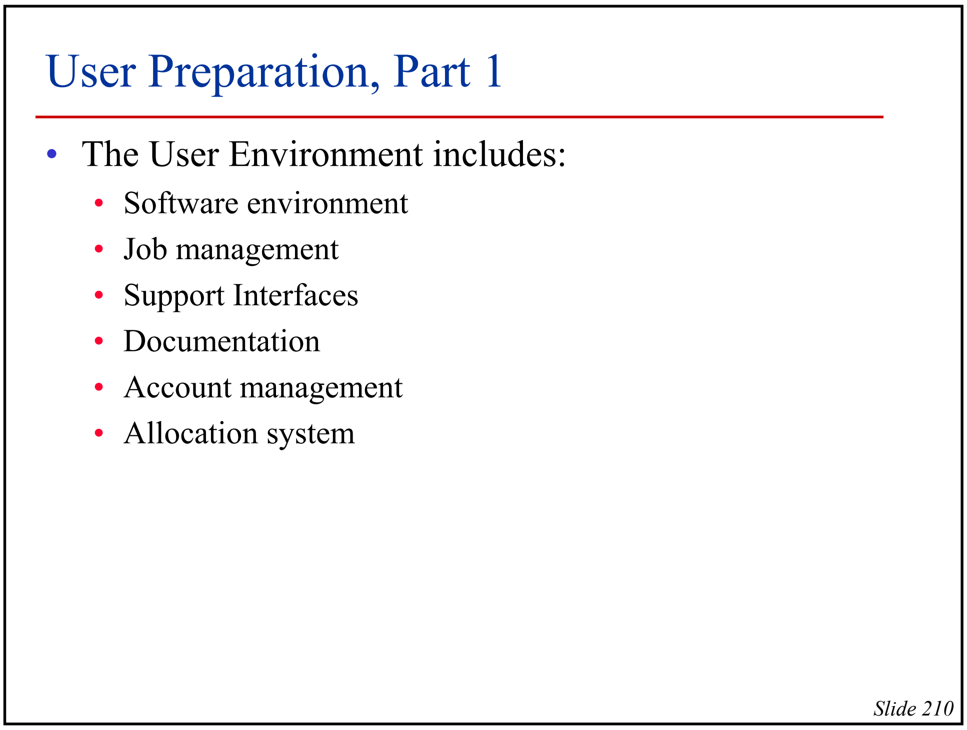
Slide 208



Final Check

- Finally, are there any lingering issues?
- Check through earlier to-do lists.
- Check the Go/No Go production checklist.
- Sleep on it.
  - If you can sleep, things are probably fine.

Slide 209



User Preparation, Part 1

- The User Environment includes:
  - Software environment
  - Job management
  - Support Interfaces
  - Documentation
  - Account management
  - Allocation system

Slide 210

Getting the user environment right

- A good software environment is critical for cluster usability.
- Software choices and some tools were discussed in detail in the **Application and Environment** section. We'll summarize here and cover a few additional points.
- The key to getting it right is making user configuration of their personal environment easy. You need something more than editing dot files with vi or emacs.
- As mentioned before, you need to support coexistence of multiple versions of the software and provide a way to switch between them.
- Be aware of potential conflicts between different packages. Jazz uses the 'Softenv' tool to handle conflicting FlexLM license manager LM_LICENSE_FILE values.

The software environment, continued

- Compilers are important and you will find that user expectations are high. At a minimum, you should have the GNU and Intel compilers. 3rd party software may dictate others.
- Debugging is arguably the most difficult task facing users of a distributed computing environment. You should provide as many tools as possible. Certainly GDB, DDD and Jumpshot (for MPICH users).
- Your choice of 3rd party software and libraries will depend heavily on the applications that your users will be running.
- Many users will require commercial packages such as STAR-CD or Fluent. These packages can cost significant amounts of money. In addition, they can often have their own requirements (STAR-CD requires the Absoft compiler).

Job Management
Scheduler and access policies

- Configuring your job scheduler to use a complex algorithm for prioritizing jobs can be a nightmare. Whether it is just difficult or impossible will depend on your choice of resource manager and scheduler. For possible choices see the **Resource Management** section.
- We recommend that you start with the default scheduler policies (typically includes FIFO combined with backfill) and observe.
- If there is contention for the machine, consider assigning priorities to projects or using a 'fair-share' algorithm. Fair-share tracks usage of resources (specified by you) and adjusts job priorities based on past usage of the machine.
- Confirm that node access policies (i.e. limiting compute node access to scheduled users) are implemented correctly across the cluster.
- Allocate certain systems for special use such as interactive development and testing. Chosen nodes should be representative of the cluster (i.e. same high performance interconnect, same software environment, etc).

Slide 213

User Support Interfaces

- The quality of your user support will strongly color the user's perception of the cluster. The personal touch often makes all the difference. Consider having a 'Help Desk' staffed by a real person during business hours.
- We strongly recommend that you use a trouble ticket system such as Bugzilla, RT, or (for $$$) Remedy.

Slide 214

User communication systems

- Communication with the users cover different situations:
  - **Emergency, temporal notifications** (i.e. the system has crashed, we are working on the problem; the system will be down for two hours tomorrow for hardware repair)
  - **Announcements** (user group meetings, tutorials, large reservations)
- Consider using different communication paths for the different situations:
  - Membership optional mailing list for emergency notifications, users can join when they are actively using the system, and leave when not.
  - Mandatory mailing list for all users for long term, critical issues (i.e. the cluster is reserved for SC demos the first two weeks of November).
- People operate differently, so you need to provide multiple communication paths. Almost everyone reads email, so we recommend that you use it in combination with other methods such as web status pages, or news programs.
- Be consistent. Use the same formatting, the same communication paths for the same category of notifications.

Slide 215

User Resources

- Provide comprehensive documentation:
  - Getting started guide (getting an account, running jobs, getting help)
  - Application documentation (examples, site specific information, user guides)
  - Tutorials (compiling programs, running jobs, using the debuggers, etc.)
  - System specific data (configuration information, access policies, etc.)
  - Searchable FAQs
  - Don’t forget the obvious: man pages, info files, etc.
  - If you provide a web status page, keep it up-to-date!
- Consider providing Applications Engineers to help users effectively use the cluster – parallel programming paradigms, optimization, debugging problems, etc.

Slide 216

Account Management

- Provide a simple, easy way for users to request accounts:
  - Web forms backed by a database system work well.
  - You will need to roll your own – we know of no general packages for this.
- Creating user accounts should be automated. This can be complex with approval mechanisms, security paperwork, etc., or a simple script that takes the user information and generates the password entry, home directory, etc.
- If your organization requires paper trails, the account system should provide this for you.
- If you will be implementing allocations, the account system should be integrated with the allocation management system.

Slide 217

Allocation Management

- If you need to divide the machine amoung individual users or projects over time, allocation management is probably your best option.
- Enforcing allocations requires integration with the resource manager. This is a hard problem and will require some amount of time and expertise to implement. The **Resource Management** section discusses options.
- Users will need a mechanism to request time on the system.
- Users will need a way to track their allocations, charge jobs to projects, etc.
- It is nice if you can integrate this into the user account system.

- If possible, avoid doing allocation management.

Slide 218

User Preparation, Part 2
Early User Mode

- Set user expectations at the very start
  - 50% down time during early testing is a feasible expectation
  - Large changes are likely
- Consider breaking the early user phase into two parts.
- Phase 1: The first wave of users, those that are:
  - Willing to live with major changes and instability
  - Capable of solving simple problems on their own
  - Ready to go with known working cluster applications
  - Patient
- Phase 2: The second wave of users, use them to:
  - Check ease of use
  - Locate holes in the documentation
  - Verify fixes put in place in phase 1 are correct
- Meet with the early users – perhaps at the midway point.

Slide 219

Shifting Into Production Mode

- Opening ceremonies and press releases.

- What you can expect the first couple of months:
  - Small adjustments will continue to be needed.
  - New problems will arise.
  - Lots of user trouble tickets, the majority of which should be answerable with pointers to documentation, or by writing documentation you forgot.
  - Status meetings will continue to be important.

- By the third month, if there are no major problems with the system, users should be settling down.

Slide 220

Ongoing operations

- Don’t expect the workload to lighten once the cluster has settled into full production mode.
- However, the focus should shift from a specific list to support of the user community, the system, and striving to achieve the goal of the system.
- Your time will be spent:
  - supporting users,
  - responding to emergencies,
  - tuning and upgrading hardware and software (carefully!),
  - regular system administration tasks,
  - generating additional documentation (never-ending),
  - other, site-specific tasks,
  - preparing reports justifying existence and supporting the need for more money.

Slide 221

Staffing

- Traditional staffing types
  - Systems administrators – manage the system services, software, …
  - Application engineers / consultants – help with user code issues, training
  - Operators – handle day-to-day tasks
  - Grad students – all (or none) of the above

- The amount and type of staffing you need depends on the service level you want to provide.
  - N = f(users) * f(applications) * f(cluster_size) * f(support_quality) * f(SLA)
  - 24x7 response adds a lot of operators

- Ranges of staffing for production support:
  - A small scale cluster (i.e. department) may have ½ to 3 people.
  - A larger (i.e. organization-wide) cluster may have 2 to 10 people.
  - A national production facility may have 10 to 20 people.
  - Numbers will vary based on team expertise, projects, other facilities.

Slide 222

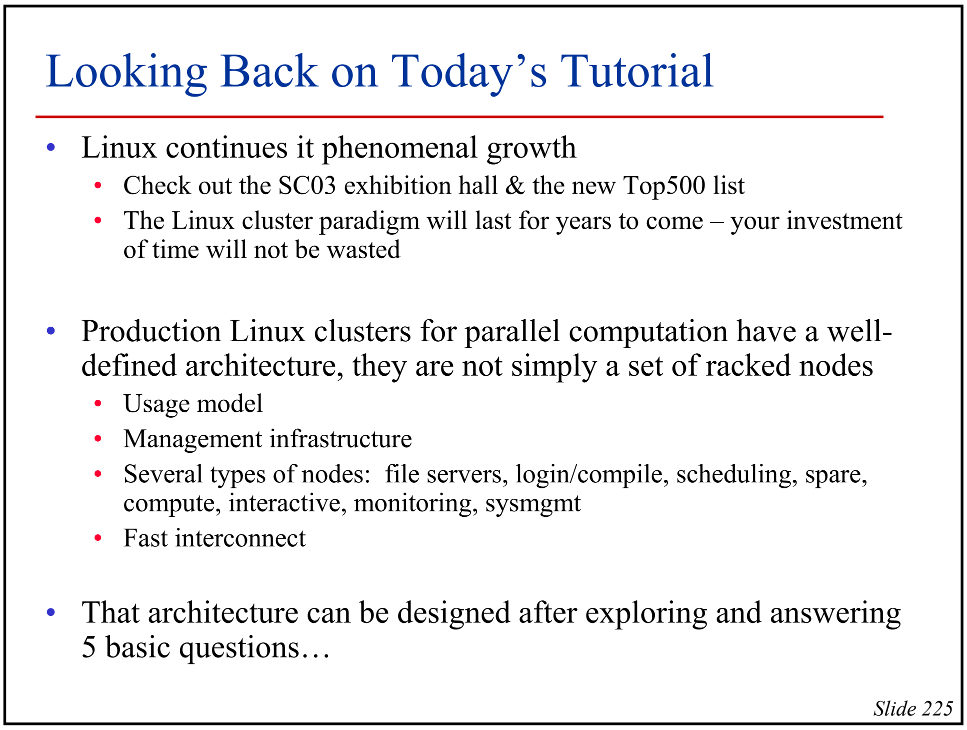
Future planning

- Expect to be doing continuous evaluations of the state of the system and planning for improvements.
- If you track problems, changes, user requests, you can use the data as inputs to the planning process for future improvements.
- Remember to learn from experiences and plan for the future changes to the system or for future systems.

Slide 223

Collecting It All Together

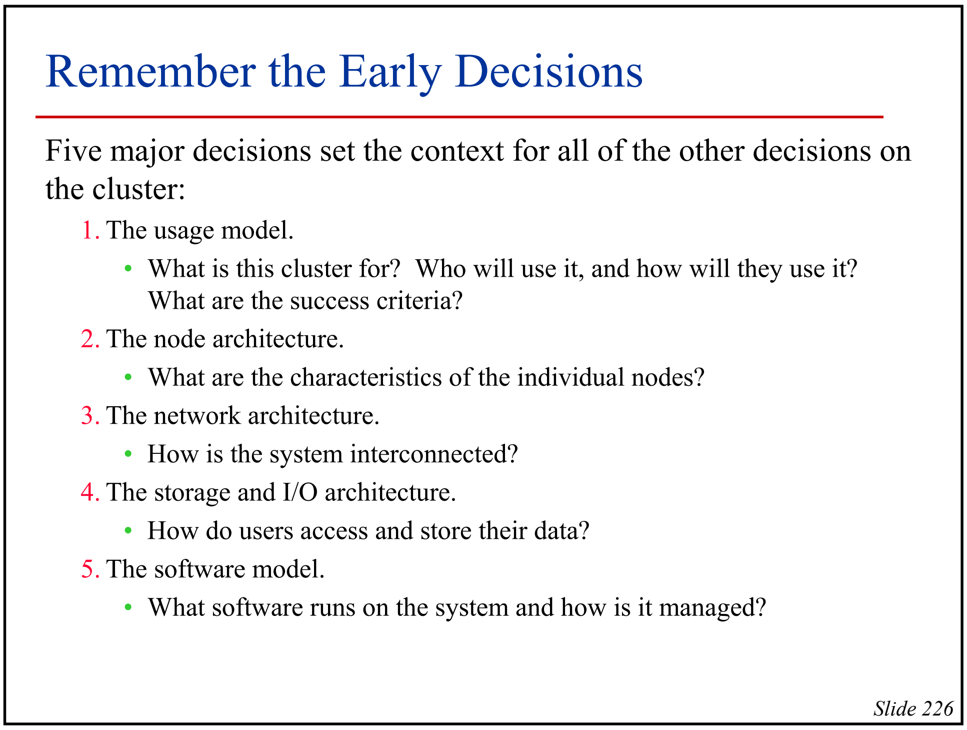
Slide 224



Looking Back on Today's Tutorial

- Linux continues it phenomenal growth
  - Check out the SC03 exhibition hall & the new Top500 list
  - The Linux cluster paradigm will last for years to come – your investment of time will not be wasted
- Production Linux clusters for parallel computation have a well-defined architecture, they are not simply a set of racked nodes
  - Usage model
  - Management infrastructure
  - Several types of nodes: file servers, login/compile, scheduling, spare, compute, interactive, monitoring, sysmgmt
  - Fast interconnect
- That architecture can be designed after exploring and answering 5 basic questions…

Slide 225



Remember the Early Decisions

Five major decisions set the context for all of the other decisions on the cluster:

1. The usage model.
   - What is this cluster for? Who will use it, and how will they use it? What are the success criteria?
2. The node architecture.
   - What are the characteristics of the individual nodes?
3. The network architecture.
   - How is the system interconnected?
4. The storage and I/O architecture.
   - How do users access and store their data?
5. The software model.
   - What software runs on the system and how is it managed?

Slide 226

Today's Most Important Message: Planning

- With careful planning production Linux clusters for serious computing can be designed, built, and operated.
- This tutorial has a wealth of information – everything from the ribbon cable to the ribbon cutting, but it all starts with planning and design.
  - Get a lab notebook, draw pictures, put your documents in a CVS repo and build a library of design and information
- The "Production Cluster Construction Checklist" (see your Tutorial Notes) can be an excellent guide.

Slide 227

The Production Cluster Construction Checklist

- Planning
  - Configuration Decisions
  - Requirements Gathering
  - Physical Constraints
  - Key Configuration Decisions
  - Management Approach
  - Vendor Input
- Arrival
  - Installation Planning
  - Shipping
  - Inventory
- Installation
  - Physical Installation
  - Initial Configuration
    - Base Networking Configurations
    - Hardware Control
    - Develop Minimal Base Image
    - Boot Process
    - Getting Images Out to All Computers
    - Remote Execution
  - System Check
    - Hardware Inventory and Testing
    - Software Inventory

Slide 228

…continued

- Configuration
  - Image Management
    - Grok the Configuration System
    - Define and Distribute Images
  - Basic Cluster Functions
    - Networks
    - Standard System Services
    - Configure File Systems
    - Cluster-specific Services
- System Testing
  - Outage Tests
  - Component Tests
  - Benchmarking
  - Stress Tests
  - Application Tests
  - Acceptance Test

Slide 229

…continued

- Preproduction
  - Timeline Check
  - Sanity Check
  - Final Configurations
  - Preparation for Users
    - Software for Users
    - Scheduler and Access Policies
    - User Documentation
    - Account Management System
    - Allocation System
    - User Communication System
    - Trouble Reporting System
  - Early User Mode
- Production
  - Announce Production
  - Ongoing Operation
  - Future Planning

Slide 230

Homework Assignment

- You paid money for a tutorial and we are assigning homework?
- Go see the technical talk "Performance Comparison of MPI Implementations over InfiniBand, Myrinet and Quadrics".
  - We have read the paper, and it is an excellent discussion of the three technologies and their performance characteristics.
- In a couple days, you will have the perfect opportunity to explore the exhibition hall and gather information.
- Use your checklists and the following questions to explore the hardware and software solutions…

Slide 231

SC03 Exhibition Scavenger Hunt (1/3) Hardware Vendors

- Find hardware nodes with good management features
  - Serial console?
  - Linux capable BIOS updates
  - Modify boot order (netboot or hard drive) from Linux or remote console
  - Configuration of RAID devices from Linux
  - Collective operations, from Linux for:
    - Power up/down
    - Reboot
  - Fault detection: can you read RAS info, ecc, etc from Linux?
  - Out of band management CPU
- Run away from web interfaces to single nodes
- How many systems have they fielded? Can you get a customer list? Where is their largest cluster?
- Does hardware come with a mass deployment tool?
- What is the STREAM bandwidth per dollar for the compute node?

Slide 232

SC03 Exhibition Scavenger Hunt (2/3) Interconnect Vendors

- Ask vendor for ratio of PCI bandwidth to/from card to link speed
- What is the **total** cost for a 1024 node machine, including all NICs, switches, and licensing?
- What tools are provided for detecting faults in the interconnect?
- Is there an option for end-to-end message error checking?
- What tools are available for stress-testing your interconnect?
- Can all of the error counters in your cluster be cleared and probed easily?
- Do they host a public user forum that you can review?
- Over the last 3 years, how often have they released a new version of their NIC?
- What is the maximum all-to-all aggregate bandwidth per dollar on a 1024-node cluster? What about just a 2 node cluster?

Slide 233

SC03 Exhibition Scavenger Hunt (3/3) Software Vendors

- Find the companies producing compilers and math libraries
  - Is there a per-node fee for using the run-time (like old HPF compilers)?
  - What are the typical speedups compared to gcc?
  - Is the compiler optimized for the AMD-64?
- How much does a debugger for a 1024 node cluster cost?
- How much does a commercial MPI for a 1024 node cluster cost?
- Are the software tools compiled for AMD-64?
- Is there a per-node fee for Red Hat on each of your 1024 nodes?
- How much is a job scheduler for 1024 nodes?
  - What features does it have compared to the competitors?
- Are there any tools for accounting and billing management for cycles used on the cluster? What happens if a job is aborted?

Slide 234

Questions and Comments? Feedback?

- If you find cool stuff on the Exhibition Floor, please send mail to tribble03@mcs.anl.gov (the authors of this tutorial).
- Please let us know how we can improve this tutorial.
- Thanks for coming!

Slide 235

Appendix A – References

These references are listed in the order that they appear in the slides. The headings indicate the associated tutorial section.

Linux Clusters and HPC

Top 500 List
www.top500.org

Cluster Architecture

In Search of Clusters, by Gregory Pfister. Prentice Hall

Beowulf project
www.beowulf.org

MPI
www.mpi-forum.org

Node Architecture

Tom's Hardware
www.tomshardware.com

Sharky Extreme
www.sharkyextreme.com

Stream Benchmark Results
www.cs.virginia.edu/stream

SPEC Benchmark Results
www.spec.org

Bay Tech Power Controllers
www.baytechcd.com

APC Power Controllers
www.apc.com

Comtrol Rocketport Serial Concentrators
www.comtrol.com

Cyclades Serial Concentrators
www.cyclades.com

LinuxBios
www.linuxbios.org

Network Architecture

Netperf IP Benchmark
www.netperf.org

Myrinet
www.myri.com

Quadrics
www.quadrics.com

InfiniBand Vendors
www.mellanox.com
www.topspin.com
www.infinicon.com

Storage and I/O

MPI-IO specification
mpi-forum.org

Lustre
www.lustre.org

PVFS
www.parl.clemson.edu/pvfs

GFS
www.sistina.com

GPFS
www-1.ibm.com/servers/eserver/clusters/software/gpfs.html

ROMIO
www.mcs.anl.gov/romio

HPSS
www4.clearlake.ibm.com/hpss/index.jsp

Purchasing and Installation

PDSH
    www.llnl.gov/linux/pdsh/

C3
    www.csm.ornl.gov/torc/C3/

System Configuration

The Practice of System and Network Administration, by Thomas Limoncelli and Christine Hogan. Addison-Wesley

Essential System Administration, by Aeleen Frisch. O'Reilly and Associates

Cfengine
    www.iu.hioslo.no/cfengine/

System Installer
    systeminstaller.sourceforge.net

System Imager
    www.systemimager.org

xCAT
    www.alphaworks.ibm.com/tech/xCAT?open&ca=dgr-lnxw16awxCat

Clusterworx
    www.linuxnetworx.com/products/clusterworx.php

Hostbase
    www.mcs.anl.gov/systems/tools

NAGIOS
    www.nagios.org

Ganglia
    ganglia.sourceforge.net/

Clumon
    clumon.ncsa.uiuc.edu/

Supermon
    sourceforge.net/projects/supermon/

Resource Management

BProc
    www.scyld.com

P4/MPD
    www.mcs.anl.gov/mpi/mpich

PBS
    www.openpbs.com
    www.pbspro.com

Maui Scheduler
    www.sourceforge.net/projects/mauischeduler

Catalina Scheduler
    www.sdsc.edu/catalina

Sun Grid Engine
    www.sun.com/software/gridware/sge.html

Load Sharing Facility
    www.platform.com/products/LSF

Qbank
    www.emsl.pnl.gov/docs/mscf/qbank

Applications and Environment

MPI Implementations

    MPICH
        www.mcs.anl.gov/mpi/mpch

    LAM/MPI
        www.lam-mpi.org

    MPI/Pro
        www.mpi-softtech.com

Compilers

Portland Group Compilers
www.pgroup.com

Intel Compilers
www.intel.com/software/products/compilers

Scientific tools

Comprehensive list
www.ncsa.uiuc.edu/UserInfo/Resources/Software/Repository

BLAS by Kazushige Goto
www.cs.utexas.edu/users/flame/goto

BLAST
www.ncbi.nlm.nih.gov/BLAST

FFTW
www.fftw.org

PETSc
www.mcs.anl.gov/petsc/petsc-2

IMSL
www.vni.com/products/imsl

NAG
www.nag.com/numeric/numerical_libraries.asp

Debuggers

Totalview debugger
www.etnus.com

Data Display Debugger
www.gnu.org/software/ddd

Performance Profiling

Jumpshot
www.mcs.anl.gov/perfvis/software/viewers/index.htm

XMPI
www.lam-mpi.org/software/xmpi

PAPI

icl.cs.utk.edu/projects/papi

PerfSuite
perfsuite.ncsa.uiuc.edu

Tau
www.cs.uoregon.edu/research/paracomp/tau

Version Management

Softenv
www.teragrid.org/docs/softenv
Modules
modules.sourceforge.net
Switcher
www.csm.ornl.gov/oscar

HPC Distributions

LLNL CHAOS project kernel information
www.llnl.gov/linux/ucrl-jc-153559.html

OSCAR
oscar.sourceforge.net

Rocks
www.rocksclusters.org

CLIC
Clic.mandrakesoft.com

Clustermatic
www.clustermatic.org

Cplant
www.cs.sandia.gov/cplant/

Condor
www.cs.wisc.edu/condor/

Entropia
www.entropia.com

MOSIX
    www.mosix.org

OpenMOSIX
    Openmosix.sourceforge.net

Serious Production Mode

Bugzilla
    www.bugzilla.org

RT: A request tracker
    www.bestpractical.com/rt/

Remedy
    www.remedy.com

Appendix B – The Production Cluster Construction Checklist

The following document is an Argonne National Laboratory technical report created by a number of cluster system engineers at Argonne.

It is included here because it presents an organized list of the major activities involved in the construction of a production cluster, thus may provide helpful context for topics covered in the tutorial. Many of the issues mentioned in the checklist are covered in more detail in the tutorial slides.

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, IL 60439

ANL/MCS-TM-267

The Production Cluster Construction Checklist

by

Rémy Evard, Peter Beckman, Sandra Bittner, Richard Bradshaw, Susan Coghlan, Narayan Desai, Brian Finley, Eugene Rackow, John-Paul Navarro
{evard, beckman, bittner, bradshaw, smc, desai, finley, rackow, navarro}@mcs.anl.gov

Mathematics and Computer Science Division

Technical Memorandum No. 267

October 2003

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38.

Available electronically at http://www.doe.gov/bridge

Available for a processing fee to U.S. Department of Energy and its contractors, in paper, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

The Production Cluster Construction Checklist

by

Rémy Evard, Peter Beckman, Sandra Bittner, Richard Bradshaw, Susan Coghlan,
Narayan Desai, Brian Finley, Eugene Rackow, John-Paul Navarro
{evard, beckman, bittner, bradshaw, smc, desai, finley, rackow, navarro}@mcs.anl.gov

Introduction

This document is a detailed checklist of the steps that one must go through to bring up a production computing cluster. The list starts with planning activities and culminates in the activities necessary to operate and sustain a production computing facility.

This checklist is derived from a number of experiences installing real-world, large-scale clusters. While each installation experience was unique, we were interested in determining the common characteristics across each deployment. We collected all of the to-do lists, presentations, notes, email messages, white board notes, and any other planning tools we could find from each of the installation activities. We combined them into a huge, messy diagram that was probably impossible to understand without having been involved in its creation but was excellent for identifying differences and commonalities. After organizing, checking, and distilling the information, we created the checklist presented here.

Interesting is the fact that the high-level activities on the resulting list are neither cluster nor computer specific. Most of these activities would be followed when installing a production computer of any architecture or when installing any kind of complex facility that will eventually support users.

The purpose of this list is not to give step-by-step instructions but rather to serve as a guide and a reminder. The items on the list are necessarily brief statements. Detailed explanations of these would go beyond the intended scope of the list.

The list is organized in outline fashion. The major phases of construction are individual sections. Each of the subsections is a task or subtask in that phase.

The items on this list are presented in a logical sequence, in approximately the order that one would follow if one were to start with a budget and an idea. However, every cluster is different, and every situation for using clusters is different. Most likely, no one would ever follow the steps here in this exact order; many things can be done in a different order, simultaneously, or skipped altogether. The list, for example, may place more emphasis on testing than many sites formally will.

Please send recommendations for improvements to future versions of this list to Rémy Evard at evard@mcs.anl.gov.

1 Planning

Having determined that you need a cluster and have a budget, determine specifically what it is that you want and who will provide it. If working with a vendor, your goal is to have a final quote and contract.

1.1 Configuration Decisions

Develop a detailed plan for the cluster.

1.1.1 Requirements Gathering

- User expectations
- Usage model
- Success criteria

1.1.2 Physical Constraints

- Space limitations
- Power limitations
- Cooling limitations

1.1.3 Key Configuration Decisions

- Node architecture
- Network architecture
- Storage and I/O architecture
- Special devices (e.g. visualization, acquisition)

1.1.4 Management Approach

- Core administrative functionality
- Security policies
- Software model

1.2 Vendor Input

Work with vendors to define a system that meets your requirements and fits in your budget. Most likely this will be an iterative process in which you adapt your configuration based on what can be provided.

- To vendor or not to vendor – determine whether you will be working with a systems integrator or rolling your own cluster from parts.
- RFP
- Quotes
- Negotiation, including Acceptance Tests
- Contract

2 Arrival

Move from having a final quote from the vendor to having all of the correct hardware at your site, ready to be installed.

2.1 Installation Planning

- Confirmation of vendor's role in installation and configuration
- Floor plans
- Rack layout plans

- Wiring plans
- Layout logistics
- Power availability and connectivity
- Fire suppression

2.2 Shipping

- Confirmation of date and shipping logistics
- Unloading and unpacking space

2.3 Inventory

- Check shipping receipts against boxes that have arrived
- Visually inspect boxes for damage
- Check contents of boxes against final quote from vendor

3 Installation

Set up the system, and carry out initial inventory tests.

3.1 Physical Installation

Install the hardware in the racks, attach the networks and storage devices, and connect everything to the power system. If a system integrator is involved, it will do all of this but may need supervision.

3.2 Initial Configuration

Get the cluster to the point where it is on and responsive enough that you can run simple jobs across the entire cluster in order to do system checking.

3.2.1 Base Networking Configurations

- Have a network space and network name plan
- Prepare the network switch(es)
- Implement the network mechanisms in the base image or boot mechanisms

3.2.2 Hardware Control

- Set up the remote power control mechanism, if any
- Set up the remote console mechanism, if any
- Force hardware reboots on individual components to test them

3.2.3 Develop Minimal Base Image

This may be provided by the vendor, may be minimal, or may be nearly complete.

3.2.4 Boot Process

- Set up the boot mechanism
- If using DHCP, may need to collect Ethernet addresses
- Set up the boot server with appropriate images

3.2.5 Getting Images Out to All Computers

- Boot all of the nodes and servers, ensuring that they get the correct images

3.2.6 Remote Execution

- Determine how to execute commands on all systems

3.3 System Check

With the system connected and responsive enough to boot all nodes and execute programs, it is now possible to carry out initial functionality and inventory tests.

3.3.1 Hardware Inventory and testing

Confirm that each of the components has the right parts and that the system is connected properly.

- Check CPUs, memory, disk, and temperature of systems.
- Check pings across the networks.
- Check network arch and configs.
- Test storage accessibility, configuration, and size.

3.3.2 Software Inventory

When using software provided by a third party such as with prebuilt vendor systems or HPC distributions, check to see what software exists.

- Base OS, libraries, kernels, etc.
- What else is installed? Does it look sane?

4 Configuration

Carry out all of the systems administration tasks that transform the individual hardware components into a functioning cluster.

4.1 Image Management

4.1.1 Grok the Configuration System

Understand the image and configuration mechanisms. Either the system provided will have certain expectations and tools, or you will need to determine how to do this yourself.

- How do you define images?
- How do you push out images?
- How do you push out changes?
- How do you install software?
- How do you install patches?

4.1.2 Define and Distribute Images

Build the base images for all parts of the cluster, modified as you see fit.

- Primary cluster server
- Any secondary servers: login, management, other
- Compute nodes
- Any specialized nodes

4.2 Basic Cluster Functions

4.2.1 Networks

- Tune or modify the IP space and Ethernet
- Configure any other networks

4.2.2 Standard System Services

- Disable all the services you won't be using
- Configure all the services you'll be using: DNS, NTP, SSH/RSH, NIS, any others

4.2.3 Configure File Systems

- File systems on individual nodes
- Home file systems and servers
- Cluster-wide file systems and servers
- Parallel file systems and servers
- Other specialized I/O such as mass storage access

4.2.4 Cluster-specific Services

- Schedulers
- Resource managers
- Parallel tools

5 System Testing

With the cluster basically operational, carry out tests to confirm that the system is working correctly. Ensure that you have a strategy for addressing any problems that you discover.

5.1 Outage Tests

- The reboot test. Reboot the entire system, see if it comes back completely without help.
- The power lossage test. If you have the nerve, test to see what happens if you kill the power.

5.2 Component Tests

Test all individual system components in the cluster.

- File system performance tests.
- The "everybody core dump at once" test
- High availability file system tests. What happens if a server goes down?
- Network connectivity tests
- Network performance tests
- File transfer into and out of the cluster

5.3 Benchmarking

Run standard benchmarks such as LINPACK and the NAS benchmarks to test overall system performance.

5.4 Stress Tests

Run any stress tests of individual components or the overall system that you deem necessary. Typically these are tough on the system, run for a long time, or both.

5.5 Application Tests

If certain applications are essential to your user community, run these applications to test functionality, performance, and correctness.

5.6 Acceptance Test

Run acceptance tests, typically as determined during the contract process, to determine what the vendor must address before the system is accepted.

6 Preproduction

At this point, the system is known to be working. Complete the work necessary to support users.

6.1 Timeline Check

Make or confirm the plan to shift into production mode. The timeline will drive the priorities for software installation and user support.

6.2 Sanity Check

- Confirm ability to do core systems administration tasks:
    - Add and remove nodes
    - Replace hardware
    - Add, upgrade, and remove software
    - Modify and distribute images
    - Remotely manage hardware
    - Add and remove user accounts
- Confirm usage policies
- Confirm security implementation

6.3 Final Configurations

Configure and test any remaining systems.

- Monitoring systems
- Backups
- Failover services for DNS, NIS, etc
- Check all the early to-do lists for lingering issues

6.4 Preparation for Users

Install the software and systems necessary to support and manage users on the system.

6.4.1 Software for users

- Compilers, development libraries, debuggers.
- Third-party software

6.4.2 Scheduler and Access Policies

- Develop initial scheduler policies
- Confirm node access policies, i.e., that computing nodes are only available to scheduled users.
- Allocate certain systems for special use such as interactive development and testing.

6.4.3 User Documentation

- FAQs such as "how to get an account" and "how to report problems"
- Documentation specific to the system, such as hostnames, access policies, etc.
- Quick-start guides
- Tutorials
- Software documentation

6.4.4 Account Management System

- A mechanism for users to request accounts
- A mechanism for creating users accounts
- Paper trails as necessary for your organization

6.4.5 Allocation System

- If usage will be tracked, determine how users will request time on the system, how allocations will be enforced, and how usage will be monitored

6.4.6 User Communication System

- Set up systems for sending announcements to users
- Consider mailing lists for user discussion

6.4.7 Trouble Reporting System

- Set up systems for users to report problems, typically a trouble ticket system

6.5 Early User Mode

- Determine criteria for early users
- Set expectations for system availability and reliability
- Consider time (duration, schedule) for regular system downtime
- Respond to user issues
- Meet with early users

7 Production

Having completed all the advance preparation, open the system up to all appropriate users. Shift emphasis from configuration to operation. Most of the items described here will be carried out simultaneously.

7.1 Announce Production

- Choose a date to change to production.
- Hold any necessary ceremonies, announcements, etc.

7.2 Ongoing Operation

Operations mode is not oriented around a specific list as much as continued support of the user community, the system, and striving to achieve the goal of the system. With a large user community, the workload will be about the same as it was for system configuration.

- Support users
- Respond to problems
- Respond to emergencies
- Tune and upgrade the system
- Continue regular tasks
- Create documentation
- Monitor and report on usage
- Other – very much subject to site expectations

7.3 Future Planning

Remember to learn from experiences and plan for the future changes to the system or for future systems.

Appendix C – Regular Administrative Tasks

This is a list of regular administrative tasks that one carries out on a production computing facility, based on a log of actual administrative tasks performed over the course of a few weeks. The list is primarily meant to give an idea of the scope and range of work directly related to sustained operation, and does not include other aspects of typical positions such as interacting with vendors, developing tools, and so on.

Hardware

- monitor
- diagnose
- replace/repair
- evaluate potential upgrades
- document

Network

- monitor
- diagnose
- replace/repair
- evaluate potential upgrades
- document

Scheduler

- monitor
- reconfigure
- diagnose
- upgrade
- evaluate potential replacements
- document

Job management

- monitor
- diagnose
- repair
- archive
- tools

Security

- patch management
    - evaluate
    - test
    - apply
    - archive
    - document
- monitor/intrusion detection
- log management
- upgrade software
- verify/test regularly
- document

Configuration management

- monitor
- verify/test regularly
- upgrade software
- evaluate
    - changes to cluster for additions to configuration
    - potential replacements
- manage base images
- document

Automation

- evaluate
    - potential replacements
    - potential actions for additional automation
- new tasks
- verify/test regularly
- update
- document

Application Management

- install
- upgrade
- clean
- document

Account Management

- creation
- modification
- inactive
- close
- access mgt
- log
- automation

Accounting Management

- monitor
- log usage
- reports
- tools

User Support

- document
- diagnose
- repair
- trouble ticket mgt
- provide information
    - mail
    - web
    - online
    - event notification

Mail System

- system mail management
  - monitor
  - diagnose
  - archive
  - repair
- user mail management
  - monitor
  - diagnose
  - repair
  - create
  - maintain
  - delete
- mail list management
  - monitor
  - archive
  - access mgt
  - document